Research in Engineering Design (2023) 34:39–60
https://doi.org/10.1007/s00163-022-00394-z

ORIGINAL PAPER

Managing multi-goal design problems using adaptive leveling-weighting-clustering algorithm

Lin Guo[1] · Jelena Milisavljevic-Syed[2] · Ru Wang[3] · Yu Huang[3] · Janet K. Allen[4] · Farrokh Mistree[5]

Received: 19 November 2021 / Revised: 5 June 2022 / Accepted: 29 July 2022 / Published online: 25 September 2022
© The Author(s) 2022

Abstract

In this paper, we address the issue of solving problems with multiple components, multiple objectives, and target values for each objective. There are limitations in managing these multi-component, multi-goal problems such as the need for domain expertise to combine or prioritize the goals. In this paper, we propose a domain-independent method, Adaptive Leveling-Weighting-Clustering (ALWC), to manage the exploration of design scenarios of multi-goal, engineering-design problems. Using ALWC, designers explore combinations and priorities of the goals based on their interrelationships. Through iteration, design scenarios are obtained with higher goal achievements and an improved understanding of the relationship among subsystems. This is achieved without increasing computational complexity. This knowledge is helpful for multi-component design. The ALWC method is demonstrated using a thermal-system design problem.

Keywords Multi-goal problems · Compromise decision support problems · Adaptive Leveling-Weighting-Clustering (ALWC) method · Clustering analysis

List of symbols

A	The number of design scenarios
Cluster	A two-dimensional array containing goal clusters of two multi-goal scenarios
$\mathcal{D}$	Matrix of deviations
D_{ij_cor}	Elements of correlation matrix $\mathbb{I}_{\mathcal{D}}$
D_{ij_ort}	Elements of the orthogonality matrix $\mathbb{I}_{\mathcal{D}}$
d_{ak}	The deviation of Goal k in Design Scenario a
d^-	The under-achievement of a goal, $G(x)$, versus its target t. $d^- \geq 0$
d^+	The over-achievement of a goal, G(x), versus its target t. $d^+ \geq 0$. Since in one scenario, a goal should either over-achievement its target, or under-achievement its target, otherwise exactly achievement its target, so, $d^-_{ak} \cdot d^+_{ak} = 0$
DS	Design scenario
$\mathcal{F}$	Feasible design space
$G_k(x)$	Specific goal k
$\mathbb{I}_{\mathcal{D}}$	Interrelationship matrix of deviations. In this paper, we explore two types of interrelationships, the correlation and orthogonality as a prototype. There can be more types of interrelationships
K	Number of goals in a problem
k	A specific goal $k = 1, \ldots, K$
κ	Number of levels of K goals in a certain design scenario
O	In a two-dimensional graph of goal deviations of 2 goals, the utopia point is where the deviation of both goals is zero
$\overline{OI}$	The diagonal of the goal space

✉ Janet K. Allen
janet.allen@ou.edu

[1] Department of Industrial Engineering, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA

[2] School of Aerospace, Transportation and Manufacturing, Cranfield University, Cranfield MK43 0AL Liverpool, UK

[3] School of Mechanical Engineering, Beijing Institute of Technology, 5 Zhongguancun South Street, Haidian District, Beijing 100081, China

[4] Systems Realization Laboratory, School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK 73019, USA

[5] Systems Realization Laboratory, School of Aerospace and Mechanical Engineering, University of Oklahoma, Norman, OK 73019, USA

$\mathcal{P}$	The set of parameters of a cDSP, including coefficients and right-hand side values of all equations
t_k	The target value of Goal k, so the equation to represent Goal k is $\frac{G_k(x)}{t_k} + d_k^- - d_k^+ = 1$
x_a^s	The satisficing solution in the design scenario corresponding to $\varkappa_{DS_a}(d)$
$\varkappa(d)$	A combination of goals.
$\varkappa_{DS_a}(d)$	Merit function of the goals for design scenario A
α_{ij}	Indicates the correlation between Goal i and Goal j
$\alpha_{ij}(\mathrm{DS}_\kappa, \mathrm{DS}_t)$	The angle between $\overline{D_\kappa^{ij} D_t^{ij}}$

1 Frame of reference

In engineering design, there is a need to infuse the knowledge of downstream activities into the design process to iteratively improve model formulation and approximation.

1.1 Features of engineering-design problems

A designer concurrently needs to consider the interactions among subsystems and tradeoffs among their conflicting requirements while simultaneously meeting multidisciplinary constraints, such as cost, physical properties, manufacturing capacity, etc. (Wang 1994). This means that the goals may have different units, and therefore, the evaluation of the tradeoffs can be difficult. Hence, *satisficing* solutions are desired; these solutions meet multiple design requirements and are relatively insensitive to uncertainties (Wang et al. 2018).

1.2 Studies and gaps on multi-component, multi-goal problems

1.2.1 Regarding problem formulation

Previati and coauthors (2019) use topology to allocate the shared part of the design space to each component and identify the optimal structural layout while maintaining its original geometry. You and Smith (2016) use atomic theory and fuzzy clustering to configure modules and manage multiple design objectives. Chiu and coauthors (2021) use this concept of Design for Assembly to model a packaging system as a three-objective problem combining integer programming and goal programming.

Regarding solution algorithms: (Fishburn 1974) review the literature in Lexicographic methods and categorize them in two trends, mathematical and pragmatic. However, while applying Lexicographic methods is that the utility theory based on human preferences are certain, descriptive, and rational. Therefore, the definition of "best" alternative depends on the context and evolve during the design. Avigad and Matalon (2011) solve multiple single-objective problems with shared components using an evolutionary algorithm. The authors compare various approaches to search for common components of multiple systems or modeling objects. Katsikopoulos (2011) explores the psychological heuristics (as an alternative to optimization) in making inferences and finds out it has mathematical basis and advantages comparing with optimization when making decisions about the future. Meier et al. (2016) explore particular subsets of the Pareto-optimal solutions for time–cost tradeoffs in product development using managerial work policies. Moon et al. (2014) use multi-objective particle swarm optimization (MOPSO) to determine variables for optimal platform strategy for designing a product family.

In this paper, we categorize multi-component or multi-goal problems into two categories—focusing on improving the performance of the solution algorithms and/or design improvement. In the papers on solution algorithms, the main focus is on sorting near-Pareto frontiers (Deb et al. 2002; Seada and Deb 2014). The performance of an algorithm is evaluated by criteria such as solution optimality, diversity of solutions, and the computational complexity of the algorithm (Soltani et al. 2002). Whereas in the papers on design improvement, the authors focus on improving the model formulation or acquiring more knowledge about the interrelationships among the components (Tang et al. 2010). Some typical methods in both categories are shown in Table 1. In our paper, the combination of these two approaches is addressed.

The limitations of the existing literature on multi-component, multi-goal problems can be summarized as follows.

- *Searching the near-Pareto frontier of optimal solutions that are sensitive to model errors and variations*. Optimal solutions are sensitive, because a problem needs to meet the necessary and sufficient Karush–Kuhn–Tucker (KKT) conditions at an optimal solution point, and the probability that a model error or variation violates the KKT conditions is relatively high. Another type of solution, a *satisficing* solution, only requires the problem to meet the necessary KKT condition. Thus, these solutions are relatively insensitive to model errors and variations. Numerical comparisons between optimal solutions and *satisficing* solutions are given in Chapter 2 of (Guo 2021).
- *Lack of decision support about using solutions in different areas of the design space in various situations*. When there are too many choices but little knowledge on how to select them, decision-makers may be overwhelmed and end up with wrong decisions (Toffler 1970).

Table 1 The features and limitations of some classic multi-objective (multi-goal) solution algorithms and methods

Emphases	Algorithm or method	Description	Searching for interior solutions	Having adaptable ways of producing more solutions (offspring)	Giving information of tradeoffs among multiple characteristics (such as solution accuracy vs. convergence speed)	Use the knowledge of tradeoffs among goals (objectives) to improve model formulation	Explore multiple forms of goals (objectives) structure, such as decomposing the problem, reducing goals, etc
Solution algorithms	VEGA (Schaffer 1985)	Using vector-valued feedback with adaptive procedures for searching high-order multi-objective problems		*			*
	SPEA2 (Zitzler et al. 2001)	Using fitness assignment, archiving, and truncating (the near-Pareto frontier) to evolve solutions to approach the Pareto-optimal set	*	*	*		
	MOEA/D (Zhang and Li 2007)	Decomposing a problem into scalar optimization sub-problems and optimizing them simultaneously	*	*	*		*
	NSGA-II/III (Deb et al. 2002; Seada and Deb 2014)	Using nondominated sorting evolutionary algorithm to update reference points adaptively to approach the Pareto frontier	*	*			*
	REDGA (Jaimes et al. 2009)	Reducing the number of objectives by removing redundant (to some degree) objectives			*	*	*
	IC-MOP (Avigad and Moshaiov 2009)	Taking into account decision makers' preferences as well as computational performances	*	*	*		
	HypE (Bader and Zitzler 2011)	Using Monte Carlo simulation to approximate the exact hypervolume values and seeking ranking of solutions	*	*	*		
	Multi-Level Decisions (Mistree et al. 1994)	Using two design strategies and multi-level decisions to foster discussion on multi-objective problems			*	*	

Table 1 (continued)

Emphases	Algorithm or method	Description	Searching for interior solutions	Having adaptable ways of producing more solutions (offspring)	Giving information of tradeoffs among multiple characteristics (such as solution accuracy vs. convergence speed)	Use the knowledge of tradeoffs among goals (objectives) to improve model formulation	Explore multiple forms of goals (objectives) structure, such as decomposing the problem, reducing goals, etc
Design improvement	RCEM (Chen et al. 1997; Choi et al. 2005)	Improving the robustness of the design using indices based on the results of exploring the solution space				*	*
	Interval analysis (Hao and Merlet 2005)	Using parallel robots based on interval analysis to determine geometries to satisfy all compulsory requirements				*	*
	SOS (Avigad and Moshaiov 2009)	Based on mathematical metaphors, composed of objective and subjective components; allowing varying degrees of precision in modeling; decomposing a problem into sub-problems, and using highly simplified evaluations		*		*	*
	Level diagrams (Reynoso-Meza et al. 2013)	Comparing multiple Pareto frontiers based on different design concepts using level diagrams to support decision-making on design concept selection	*	*			*
	XPLORE in DSIDES (Smith et al. 2015a; Sabeghi et al. 2016)	Exploring the design space by exploring different goal structures using a compromise Decision Support Problem			*	*	*
	CORTHOG (Warwick 2019)	Removing poor measurement degrees-of-freedom iteratively until pseudo-orthogonality check was optimized	*	*		*	*

- *Relying on domain knowledge to combine or prioritize multiple objectives* (Smith et al. 2015), decompose the problem (Zhang and Li 2007), or set targets for the goals (Sabeghi et al. 2016). For some problems, domain knowledge may not be correct or for a new design, domain knowledge may be not be available

Given these limitations, in this paper, we want to find a domain-independent method to capture and reuse the knowledge of a multi-goal engineering-design problem to facilitate the exploration and selection of design scenarios.

To answer this, data analysis is used to obtain insight for decision support for combining goals, especially when domain knowledge is insufficient. We hypothesize that by exploring goal combinations based on their interrelationships, the rate of achievement of each goal can be improved, is more understandable, and diverse design scenarios can be identified. The process of knowledge discovery and reuse is then domain-independent. In this paper, goals are defined as objectives with target values and the design problem is managed by minimizing the deviation between what is achievable and the goal target. Therefore, design improvement is obtained by exploring the interrelationships among the system components, and *satisficing* solutions are obtained by combining the Pre-emptive and Archimedean (weighted sums) strategies of reaching goals.

In Sect. 2, we propose a knowledge-driven method, the adaptive leveling-weighting-clustering (ALWC) algorithm, to manage knowledge capture and reuse in multi-goal problems. In Sect. 3, we use a thermal-system design test problem to demonstrate the utility of the ALWC method. In Sect. 4, the results and discussion are presented. Finally, in Sect. 5, the contributions in this paper are summarized and the scope of applicability of the ALWC method is described.

2 Proposed method: the adaptive leveling-weighting-clustering (ALWC) method

We propose the adaptive leveling-weighting-clustering algorithm (ALWC) to fill the research gaps. How do we fill the research gaps using the ALWC?

First, instead of searching for a near-Pareto frontier consisting of optimal solutions that are sensitive to model errors and variations that violate the second-order KKT conditions, using the ALWC, designers obtain *satisficing* solutions that are relatively insensitive to model errors and uncertainties. By applying the ALWC, designers obtain good enough solutions that satisfy the first-order KKT conditions, but may not satisfy the second-order KKT conditions (Guo 2021).

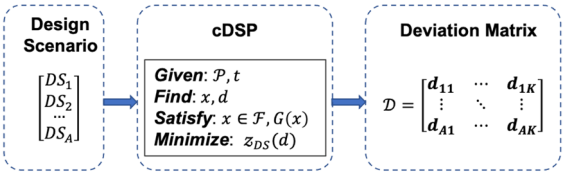


Fig. 1 Using multiple design scenarios to obtain a deviation matrix

Second, in each iteration, decision support is provided based on the awareness of subsystems and reorganizing them by changing goal combinations.

Third, instead of relying on domain expertise, using the ALWC, designers can use calculations and analyses to manage the multiple components of a system.

We explain how we define and learn the interrelationships among goals in Sect. 2.1, then we introduce how to use the interrelationships to generate and improve goal combinations in Sect. 2.2, and then describe the algorithms in the ALWC in detail in the Appendix.

2.1 Clustering the goals based on their interrelationships

In managing a multi-goal design problem to explore effective combinations of the goals $\varkappa(d)$, the interrelationships among the goals, for example, correlations or orthogonality, must be determined. Then, the scalarization function or priority of the goals is determined based on their interrelationships. A scenario, a specific combination of the goals in a multi-goal problem, is defined as a design scenario (DS), that is, an Archimedean approach using a weight vector, or a Pre-emptive approach to prioritize the goals, or a combination of the two strategies. In a design scenario, DS_a, the merit function is denoted as $\varkappa_{\mathrm{DS}_a}(d)$, the corresponding *satisficing* solution is x_a^s, and the deviation of Goal k is d_{ak}. For a K-goal problem, if we solve it using A design scenarios, $\left[\mathrm{DS}_1, \mathrm{DS}_2, \ldots \mathrm{DS}_A\right]^T$, where A is a positive integer, we get an $A \times K$ matrix of deviations, $\mathcal{D}$; Fig. 1. In the cDSP formulation in Fig. 1, $\mathcal{P}$ is the set of parameters, t is the vector of the target of the goals, x is the vector of the variables, d is the set of deviations of the goals from their targets, $\mathcal{F}$ is the feasible area, $G(x)$ is the set of goals, and $\varkappa_{\mathrm{DS}}(d)$ is the merit function of the cDSP either using Archimedean or Pre-emptive, or mixed strategy to manage the goals. The relationships that we learn by analyzing Matrix $\mathcal{D}$ are an indication of the conditional correlation or conditional orthogonality.[1] Within

[1] Correlation analysis and orthogonality analysis are two examples of analyses that can be used to determine the interrelationships among goals, but other types of analyses that can be used to capture the interrelationships among goals and designers can select their own methods and customize their approach based on the characteristics of their problems.

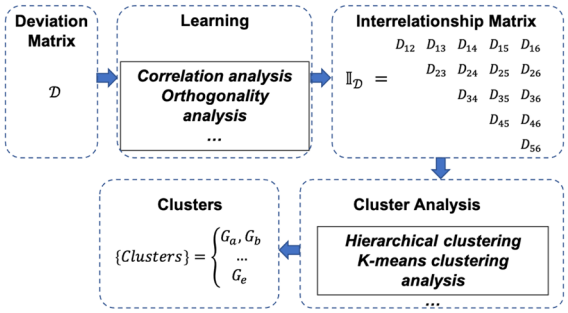


Fig. 2 Cluster analysis using a deviation matrix

the feasible space $\mathcal{F}$ and under A DSs, this is the correlation or the orthogonality of the K goals; see Fig. 2. Learning algorithms and cluster analysis methods are the used based on the problem requirements. Here, knowledge discovery and decision support related to the innovation, selection, and customization of learning algorithms and cluster analysis methods are not our focus; however, they can be addressed and improved using the ALWC method.

2.1.1 The adaptiveness

To make the learning algorithm generic, we assume no domain knowledge about the interrelationships among the goals. Here, in each iteration, only a limited number of design scenarios are generated, and we update and learn about the interrelationships through iteration. In this sense, our method is adaptive.

2.1.2 Learning the correlations

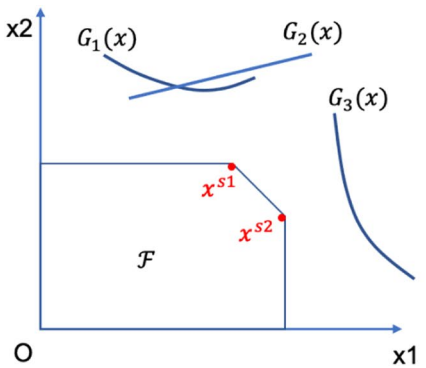
We learn the correlations from the deviation vector of the goals using angle-based distances. Suppose for a three-goal problem, we use two design scenarios DS1 and DS2 to learn about the goal correlation, and the scalarization of each design scenario is $\precsim_{DS_i}(d), d = [d_k^-, d_k^+]^T$, where $i = 1, 2, k = 1, 2, 3$. The satisficing solution and the deviations are represented in Eqs. 1 and 2

$$x^{si} = \arg\left(\min\left(\precsim_{DS_i}(d)\right)\right), \quad \forall i = 1, 2, \tag{1}$$

$$d_{ik} = \left|t_k - G_k\left(x^{si}\right)\right|, \quad \forall i = 1, 2, \forall k = 1, 2, 3. \tag{2}$$

In Fig. 3, we use a two-dimensional solution space to illustrate the three goals $G_k(x)$, $\forall k = 1, 2, 3$, and two satisficing solutions $x^{si}, \forall i = 1, 2$, for two design scenarios DS$i, \forall i = 1, 2$. In Fig. 4, we show the deviation points in two-dimensional goal spaces. We normalize the deviation of all design scenarios, d_k^-, d_k^+, in the range of [0, 1], where 0 means achieving the target of a goal the and 1 means the worst case of achieving the target of the goal. The coordinate origin O is the utopia point, which means in this scenario, we achieve the target of a goal the best among all design scenarios, either over-achievement or under-achievement. On the contrary, Point I (1, 1) is the worst point, which means in this scenario, we achieve the target of the goal the worst among all design scenarios; in other words, either d^- or d^+ is the largest in this scenario. When a changing from design scenario DS1 to DS2, the deviations of Goal 1 and Goal 3 change in different directions (Fig. 4a), whereas the deviations of Goal 1 and Goal 2 change in the same direction (Fig. 4b). Therefore Goal 1 and Goal 3 are less correlated, so they belong to two different clusters, while Goal 1 and Goal 2 have a higher correlation, so they belong to a single cluster. Suppose we use i and j to represent two goals – Goal i and Goal j, the acute angle between $D_1^{ij}D_2^{ij}$ and the diagonal $\overline{OI}$ of the goal space, α_{ij}, indicates the correlation between Goal i and Goal j, so we use an angle-based correlation analysis to understand the interrelationship between the two goals (Eq. 3). As more design scenarios are used, the correlation among goals becomes clearer. For the, A, number of design scenarios

$\alpha_{ij}\left(\text{DS}_\kappa, \text{DS}_\iota\right)$ is the angle between $\overline{D_\kappa^{ij}D_\iota^{ij}}$ the deviation coordinates of Goal i and Goal j for two design scenarios,



Design scenario	Solution point	Deviation		
		Goal 1	Goal 2	Goal 3
$DS1$	x^{s1}	$d_{11} = 0.7$	$d_{12} = 0.65$	$d_{13} = 0.6$
$DS2$	x^{s2}	$d_{21} = 0.4$	$d_{22} = 0.35$	$d_{23} = 0.8$

Fig. 3 The *satisficing* solutions to a three-goal cDSP under two design scenarios illustrated in a two-dimensional solution space

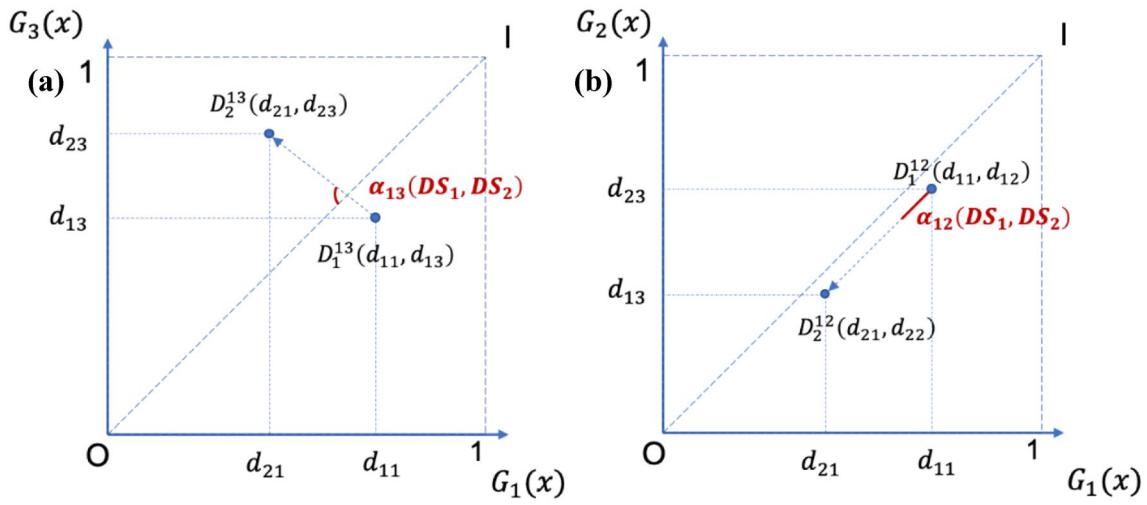


Fig. 4 The satisficing solutions to a three-goal cDSP under two design scenarios illustrated in two-dimensional goal spaces

DS_κ and DS_ι, average the angle-based correlations are shown in Eq. 4, where Cor_A^2 is the square of the correlation of the design scenario matrix.

D_{ij_cor} are elements of correlation matrix, which is a type of $\mathbb{I}_{\mathcal{D}}$ we use in this paper.

Therefore, we obtain a matrix using angle-based correlation with A design scenarios as in Eq. 5. As we update the design scenarios at each iteration, we update $\mathbb{I}_{\mathcal{D}}$. Using $\mathbb{I}_{\mathcal{D}}$ to cluster the goals, when the clustering results converge, the iteration is stopped

$$D_{ij_\mathrm{cor}}(\mathrm{DS}_1, \mathrm{DS}_2) = \mathrm{arc}(\sin(\alpha_{ij})), \quad \forall i, j = 0, \ldots K, \tag{3}$$

$$\begin{aligned} &D_{ij_\mathrm{cor}}(\mathrm{DS}_1, \ldots, \mathrm{DS}_A) \\ &= \frac{1}{\mathrm{Cor}_A^2} \sum_{\kappa=1}^{A-1} \sum_{\iota=\kappa+1}^{A} \mathrm{arc}(\sin(\alpha_{ij}(\mathrm{DS}_\kappa, \mathrm{DS}_\iota))), \\ &\forall i, j = 0, \ldots K, \end{aligned} \tag{4}$$

$$\mathbb{I}_{\mathcal{D}}\left(\mathrm{correlation}, \left[\mathrm{DS}_1, \ldots \mathrm{DS}_A\right]^T\right) = \left(D_{ij_\mathrm{cor}}(\mathrm{DS}_1, \ldots, \mathrm{DS}_A)\right). \tag{5}$$

2.1.3 Learning about orthogonality

When learning about the correlation among the goals, we treat the deviation vector of the goals as statistical variables and analyze the correlations among these variables. When learning about the orthogonality among the goals, we obtain the relative position of goals in geometric space, under the constraints of the feasible region $x \in \mathcal{F}$, using the dot-product of the deviation vector of any two goals, see Eq. 6. In Fig. 4a and b, we show the dot-product of the deviation vector of two goals, Goal i and Goal j, and for two design scenarios, *DS1* and *DS2*. The dot-product is calculated as in Eq. 7 for the Euclidean distance, or 2-norm, that is when $p = 2$ in Eq. 7. If under multiple design scenarios, the dot-product is zero, the deviation vectors of the two are orthogonal, so we conclude that the two goals are orthogonal under these design scenarios—one way of representing this is that the projection of the two goals onto a two-dimension space is perpendicular; see Fig. 5a. If the dot-product is relatively large, the two goals are closer to being parallel than being orthogonal; see Fig. 5b. Then, D_{ij_ort} are elements of the orthogonality matrix, which is another type of $\mathbb{I}_{\mathcal{D}}$ used in this paper; therefore, we obtain the matrix using orthogonality for A design scenarios, Eq. 8

$$\begin{aligned} &\{< \mathrm{Goal}i, \mathrm{Goal}j > | x \in \mathcal{F}\} \\ &= \int_{x \in \mathcal{F}} \overline{G_i(x)} G_j(x) dx \approx D_{ij_{\mathrm{Ort}}}(\mathrm{DS}_1, \ldots, \mathrm{DS}_A) \\ &= \left[d_{1i}, \ldots d_{Ai}\right] \cdot \left[d_{1j}, \ldots d_{Aj}\right]^T, \end{aligned} \tag{6}$$

$$\begin{aligned} D_{ij_{\mathrm{Ort}}}(\mathrm{DS}_1, \ldots, \mathrm{DS}_A) = &\left(\sum_{\kappa=1}^{A} {d_{\kappa i}}^p\right)^{\frac{1}{p}} \left(\sum_{\kappa=1}^{A} {d_{\kappa j}}^p\right)^{\frac{1}{p}} \\ &\cos(\theta_{ij}(\mathrm{DS}_1, \ldots, \mathrm{DS}_A)), \end{aligned} \tag{7}$$

$$\mathbb{I}_{\mathcal{D}}\left(\mathrm{orthogonality}, \left[DS_1, \ldots DS_A\right]^T\right) = \left(D_{ij_\mathrm{ort}}(\mathrm{DS}_1, \ldots, \mathrm{DS}_A)\right). \tag{8}$$

2.1.4 Cluster analysis based on the interrelationship matrix of the goals

In this paper, we use multiple clustering algorithms and use cross validation to ensure clustering the results rationally.

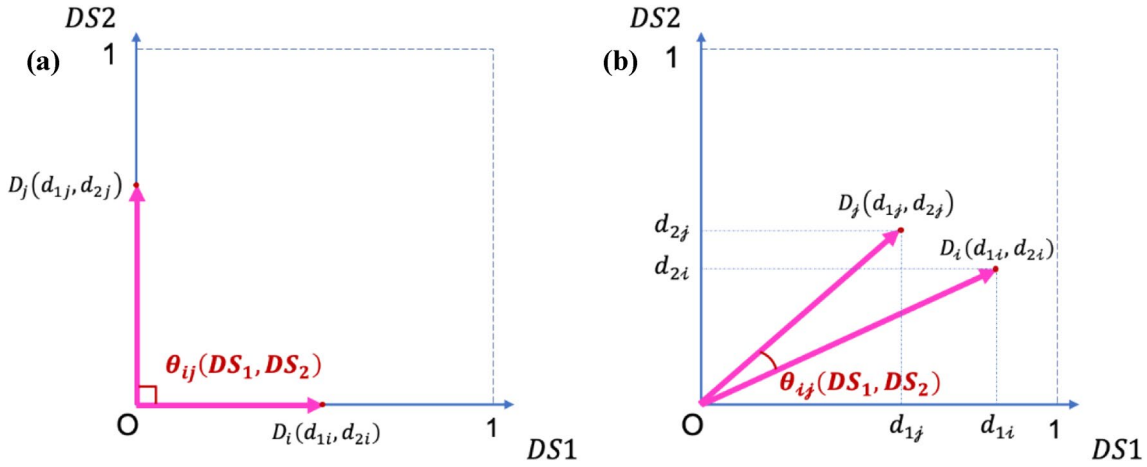


Fig. 5 The orthogonality between the deviation vectors of two goals for two design scenarios

There are criteria for designers to choose appropriate clustering algorithms, for example, the sensitivity of the clustering results to the sample size (the number of design scenarios), and the improvement of goal achievement and diversity of the solutions by leveling the goals using a clustering result. Next, we demonstrate the feasibility of further improving the selection of clustering algorithms.

2.2 A schematic of the ALWC method

The ALWC includes two loops; see Fig. 6. Leveling-Weighting-Clustering is the outer loop, and Weighting-Clustering is the inner loop. In the outer loop, leveling starts with the clustering result $\{C, \text{Cluster}\}$ from the previous iteration.[2] In the first iteration, leveling starts with the initialized single cluster and all goals are set to a single level. In later iterations, when there is more than one cluster from the previous clustering, each cluster is set to a level. After running all processes in this leveling setting, including the weighting and clustering in the inner loop, the levels are alternated, that is, the goals in each cluster are alternately set to a different level. When the inner loop is entered, the goals in each level are combined using weight vectors. After being solved using the ALP (see definition in Glossary), the deviation matrix, $\mathcal{D}$, is obtained, and the interrelationship matrix, $\mathbb{I}_{\mathcal{D}}$, is obtained and then can be used for cluster analysis. Then, a new $\{C, \text{Cluster}\}$ is returned to update leveling for the next iteration. The inner loop stops generating more weight vectors when the diversity of $\mathbb{I}_{\mathcal{D}}$ shows little increase, that is $\sigma(\mathbb{I}_{\mathcal{D}}) \le \varepsilon$, where ε is a threshold heuristically determined as more weight vectors are used. The outer loop stops leveling the goals based on the latest clustering result $\{C, \text{Cluster}\}$

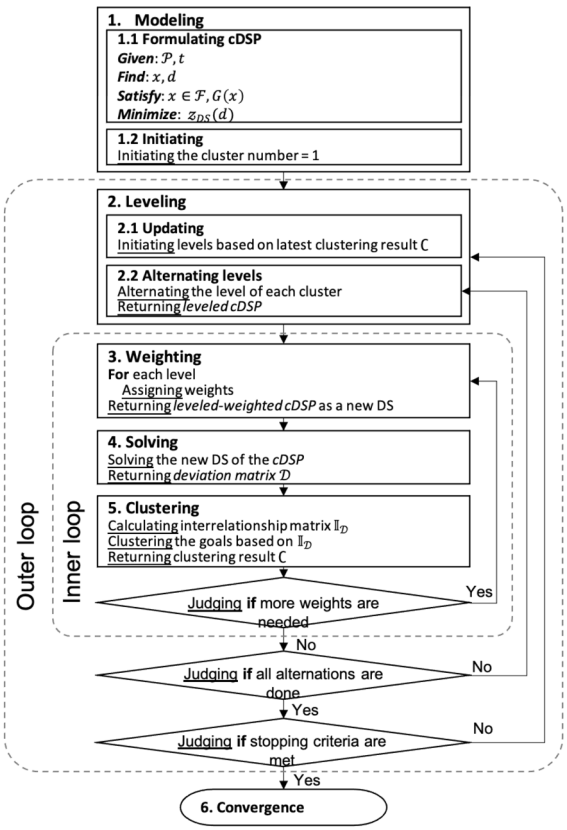


Fig. 6 The flowchart of the adaptive leveling-weighting-clustering (ALWC) loop

[2] C is the number of clusters. Cluster is a two-dimensional array containing the goal clusters, e.g., Cluster = [[G1, G2, G4], [G3, G5, G6]] indicates that there are two clusters: Cluster 1 includes Goal 1, 2, and 4, whereas Cluster 2 includes Goal 3 5, and 6. Cluster[i][j] represent the jth goal in the ith cluster, and so, Cluster[2][3] = $G6$. Array "Level" works similarly.

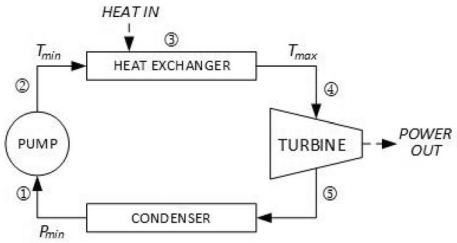


Fig. 7 The thermal system

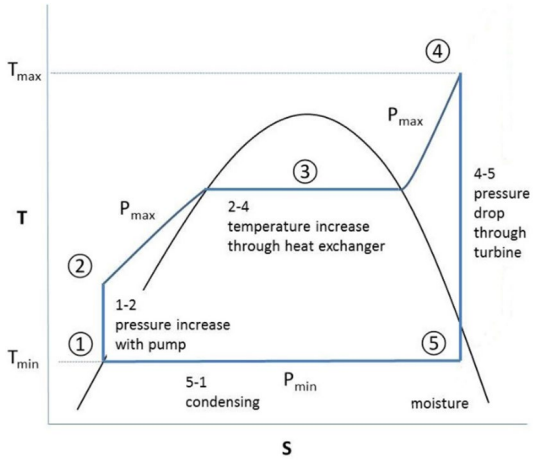


Fig. 8 Rankine cycle (temperature vs entropy)

if it has not changed in the previous η iterations, where η is a positive integer determined heuristically. Convergence is obtained when the outer loop stops,

Algorithms for Steps 2 Leveling, 3 Weighting, and 5 Clustering are given in detail in the Appendix. Steps 1 and 4 are existing methods, and for details of modeling and solution, users may refer to (Mistree et al. 1981, Mistree et al. 1993). The algorithms for calculations are in the Appendix.

We use a test problem, a thermal system design, to determine the effectiveness of the ALWC method; see Sects. 3 and 4.

3 A test problem in thermal system design

3.1 A steam-based Rankine cycle

To demonstrate the use of the ALWC method in managing a multi-goal, design problem, with an example in the design of a thermal system. The problem was first published in (Smith et al. 2015). We select this problem, because it is a multi-component, engineering-design problem with six goals. Like some other problems with nonlinear, nonconvex equations and goals with different units, when formulating the problem as an optimization model and solving it using optimization methods, we fail to identify any feasible solutions. Therefore, we demonstrate how to manage it using the cDSP as the formulation construct, the ALP as the solution algorithm (Mistree 1993), and the ALWC as the multi-goal exploration method.

There can be various applications for small scale "power" plant systems that run small generators to produce electricity or to directly use the power produced. For example, these systems may provide power to irrigation equipment, drive reverse osmosis systems for fresh water for underdeveloped areas, and generate electricity for use in small collectives.

Building a system around a steam-based Rankine cycle is a common approach for designing power plants for these small scale applications if there is an available heat source. The Rankine cycle is a theoretical representation of a heat engine that converts heat into mechanical work while undergoing phase change (Macquorn Rankine 1853; Wikipedia 2019); see Fig. 7. The major system components of the system are a turbine that produces power, a pump, a heat exchanger, and a condenser.

In this process, water is first compressed by a feed-water PUMP, and then boiled and superheated in the HEAT EXCHANGER, before being expanded through a TURBINE, which turns an electric generator, Fig. 7. The low-pressure steam is then condensed in a CONDENSER and fed back to the feed-water PUMP to be reused, Fig. 7. Thus, there are four components in the Rankine cycle thermal; it is natural for designers to treat systems independently, assuming tradeoff relationships among them solving the model or for post-solution analysis. However, relying on such domain knowledge may or may not effectively lead to robust design.

From the perspective of decision support and design improvement, such a thermal system presents complexity and dilemmas to be managed and resolved. We consider heat source issues (heat through the heat exchanger), turbine power out, and the choice of working fluids.

The performance is determined by the cycle's maximum and minimum pressures and maximum temperature (*PMAX, PMIN* and *TMAX*). Energy is transferred to the closed loop Rankine cycle through a heat exchanger. The heat exchanger is assumed to be counter flow with the key characteristic of the maximum temperature of the heating flow (*TMAXE*).

The ideal Rankine cycle involves four processes, as shown graphically in the Temperature (*T*) versus Entropy (*S*) plot in Fig. 8. The Rankine cycle consists of 4 cycles, Fig. 9:

①-② adiabatic pumping of the saturated liquid from *PMIN* to *PMAX*.

②-④ isobaric heat addition in the heat exchanger to *TMAX*,

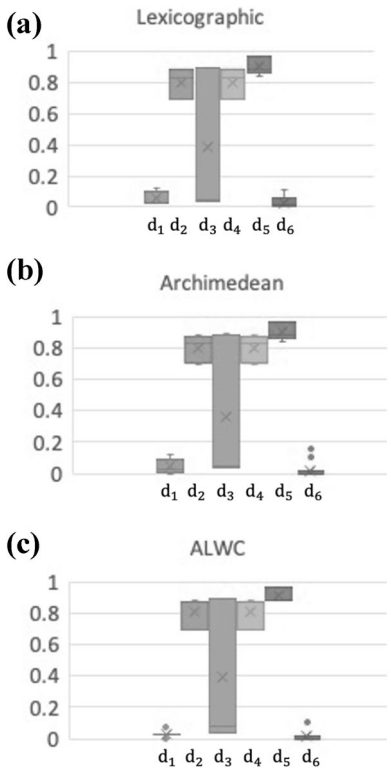


Fig. 9 The box plots of the deviations of the six goals using the Preemptive strategy, the Archimedean strategy, and the ALWC

④-⑤ adiabatic expansion in the turbine from *PMAX* to *PMIN* producing power with the possibility of wet steam exiting the turbine, and

⑤-① isobaric heat loss in the condenser.

The isothermal segments represent moving from saturated liquid to saturated vapor in the case of ③ in the heater and the reverse in the condenser between ⑤-①. The key thermodynamic properties of the working fluid(s) are determined using REFPROP (Lemmon and Huber 2013). The basic features of the problem are: four decision variables, one linear constraint, nine nonlinear inequality constraints, and six nonlinear goals.

3.2 Formulating the Rankine cycle in the compromise decision support problem

The formulation of the compromise Decision Support Problem (cDSP) is as follows.

GIVEN

Parameters including dependent system variables P1-P51 (units in abbreviated SI units)

CARNOT Carnot cycle efficiency (%) Parameter 1 (P1)

CPEE Specific heat value for input in exchanger P2

CPRE Specific heat value for Rankine (output) in exchanger (J/(kg·K)) P3

DBTMNR/DBTMNE= 273.16 P4/P5

Lower temperature limit (freezing point) in the Rankine cycle (*DBTMNR*) and in the heat exchanger (*DBTMNE*) for the fluid (K)

DBTMXR/DBTMXE= 2000.0 P6/P7

Upper temperature limit in the Rankine cycle (*DBTMXR*) or in the heat exchanger (*DBTMXE*) for every fluid (K)

DELTLM Logarithmic main temperature difference (K) P8

DENSi $i = 1, 2, \ldots 5$, Density at ①-⑤ (kg/m3) P9/P10/P11/P12/P13

EDIA/ELEN Diameter/length of heat exchanger (m) P14/P15

ENTHi $i = 1, 2, \ldots 5$, Specific enthalpy at ①-⑤ (J/kg) P16/P17/P18/P19/P20

ENTHMX/ENTHMN Enthalpy at TMINE/TMAXE in exchanger P21/P22

FLOWR/FLOWE Mass flow rate of Rankine cycle / exchanger (kg/s) P23

FRMXR The upper limit of Rankine cycle mass flow rate (kg/s) P24

HTEFF Heat transfer effectiveness (%) P25

PPUMP/PTURB Power of the pump/turbine (W) P26

PRESi $i = 1, 2, \ldots 5$, Pressure at ①-⑤ (kPa) P27/P28/P29/P30/P31

QINR Heat transfer in the heat exchanger (W) P32

QOUTE Exchanger heat transfer (W) P33

QUALi $i = 1, 2, \ldots 5$, Quality of stream at ①-⑤ (%) P34/P35/P36/P37/P38

RCEFF Rankine cycle efficiency (%) P39

RCMIT Rankine cycle moisture in turbine (%) P40

REQPOW Required power at the Rankine cycle (kW) P41

RFEEDL Calculated Rankine cycle length required given diameter (m) P42

SAREAE Surface area of the heat exchanger (m^2) P43

SYSEF1 System efficiency 1 (%) P44

STSEF2 System efficiency 2 (%) P45

TDELE=10 Requirement of minimum temperature change in the heat exchanger (K) P46

TDELC Minimum temperature gap between the minimum temperature in the heat exchanger and the temperature at ② P47

TEFFEX Temperature exchanger efficiency (%) P48

TEMPi $i = 1, 2, \ldots 5$, Temperature at ①-⑤ (K) P49

TMINE Minimum temperature in exchanger (K) P50

UHTC Overall heat transfer coefficient (W/(m^2·K)) P51

Functional relationship between parameters and system variables F1-F14

$$CARNOT = 1.0 - \frac{TEMP4}{TEMP4} \quad \text{Function 1 (F1)}$$

$$CPEE = \frac{ENTHMX - ENTHMN}{TMAXE - TMINE} \quad \text{F2}$$

$$CPRE = \frac{ENTH4 - ENTH2}{TEMP4 - TEMP2} \quad \text{F3}$$

$$DELTLM = \frac{(TMINE - TEMP2) - (TMAXE - TEMP4)}{ln(\frac{TMINE - TEMP2}{TMAXE - TEMP4})} \quad \text{F4}$$

$$FLOWE \cdot (ENTHMX - ENTHMN) = FLOWR \cdot (ENTH4 - ENTH2) \quad \text{F5}$$

$$FLOWR = \begin{cases} FRMXR & if\ EMP4 - TEMP5 = TEMP2 - TEMP1 \\ \frac{REQPOW}{(TEMP4 - TEMP5) - (TEMP2 - TEMP1)} & otherwise \end{cases} \quad \text{F6}$$

$$HTEFF = 1.0 - e^{-\frac{UHTC \cdot SAREAE}{FLOWR \cdot CPRE}} \quad \text{F7}$$

$$PPUMP = (ENTH2 - ENTH1) \cdot FLOWR \quad \text{F8}$$

$$PPUMB = (ENTH4 - ENTH5) \cdot FLOWR \quad \text{F9}$$

$$QINR = FLOWR \cdot CPRE \cdot (TEMP4 - TEMP2) \quad \text{F10}$$

$$QOUTE = FLOWE \cdot CPEE(TMAXE - TMINE) \quad \text{F11}$$

$$RCEFF = \frac{PTURB - PPUMP}{QINR} \quad \text{F12}$$

$$RCMIT = \begin{cases} 0 & if\ QUAL5 > 1 \\ 1 & if\ QUAL5 < 0 \\ 1 - QUAL5 & if\ 0 \le QUAL5 \le 1 \end{cases} \quad \text{F13}$$

$$RFEEDL = \frac{SAREAE}{\pi \cdot I} \quad \text{F14}$$

$$SAREAE = \pi \cdot EDIA \cdot ELEN \quad \text{F15}$$

$$SYSEF1 = \frac{PTURB - PPUMP}{QOUTE} \quad \text{F16}$$

$$SYSEF2 = RCEFF \cdot TEFFEX \quad \text{F17}$$

$$TEFFEX = \frac{TMAXE - TMINE}{TMAXE - TEMP2} \quad \text{F18}$$

$$UHTC = \frac{QINR}{SAREAE \cdot DELTLM} \quad \text{F19}$$

FIND

x, the decision variables (system variables) *x1-x4*

PMAX	Maximum pressure in the Rankine cycle	Variable 1 (*x1)*
PMIN	Minimum pressure in the Rankine cycle	*x2*
TMAX	Maximum temperature in the Rankine cycle	*x3*
TMAXE	Maximum temperature of the heating fluid in the exchanger	*x4*

Deviation variables *d*

d_k^-, d_k^+ $k = 1, 2, \ldots 6$, deviation variables that measure the under-achievement and over-achievement of Goal *k* versus its target

SATISFY

System constraints

Linear constraints C1

TMAXE – TMAX ≥ DELTLM Temperature delta (10 K) for maximum in exchanger, Constraint 1 C1

Nonlinear constraints C2-C10

RCMIT ≤ TMXL Moisture in turbine (RCMIT) less than upper limit (TMXL) C2

FLOWR ≤ FRMXR Rankine cycle mass flow rate (FLOWR) less than upper limit (FRMXR) C3

TEMP4 ≥ TEMP3 Temperature at ④ (TEMP4) should be greater than or equal to temperature at ③ (TEMP3) C4

$QUAL4 \geq 1.0$	Quality at ④ (QUAL4) is superheated vapor	C5
$TMAXE - TMINE \geq TDELE$	TMAXE is greater than TMINE by at least TDELE	C6
$TMINE - TMEP2 \geq TDELC$	TMINE is greater than temperature at ② by at least TDELC	C7
$CARNOT \geq SYSEF1$	Ideal Carnot cycle efficiency is greater than System efficiency 1 (Sanity check 1)	C8
$CARNOT \geq SYSEF2$	Ideal Carnot cycle efficiency is greater than system efficiency 2 (Sanity check 2)	C9
$DBTMXE \geq TMAXE$	Temperatures within valid ranges for REFPROP fluid	C10

The system variable bounds ($x_j^{min} \leq x_j \leq x_j^{max}$):

$500 \leq PMAX \leq 5000$	(kPa)	Bound 1,2 (, B1)
$350 \leq TMAX \leq 850$	(K)	B2, B3
$350 \leq TMAXE \leq 850$	(K)	B4, B5

The system goals

Goal 1: Achieve zero moisture in steam leaving the turbine (Minimize the moisture or maximize the steam quality of ①)

$RCMIT + d_1^- - d_1^+ = 0$ Goal 1 (G1)

Goal 2: Maximize Rankine cycle efficiency

$RCEFF + d_2^- - d_2^+ = 1$ G2

Goal 3: Maximize temperature exchanger efficiency

$TEFFEX + d_3^- - d_3^+ = 1$ G3

Goal 4: Maximize system efficiency indicator 1

$SYSEF1 + d_4^- - d_4^+ = 1$ G4

Goal 5: Maximize system efficiency indicator 2

$SYSEF2 + d_5^- - d_5^+ = 1$ G5

Goal 6: Maximize heat transfer effectiveness in exchanger

$HTEFF + d_6^- - d_6^+ = 1$ G6

MINIMIZE

The design scenario DS to be explored

$z_{DS}(d)$ DS

3.3 Limitations in the formulation and results in the original paper

There are limitations of the method used in Smith et al. (2015). Using the Pre-emptive approach, the six goals are placed in six priority levels. However, by prioritizing the goals differently, comparisons may show competing goals driving the solution in different directions. Using the Archimedean approach, the six goals are grouped at the same level and linearly combined using weights. There can be a mixture of the Pre-emptive and Archimedean approaches, which may organize the components based on their interrelationships, correlation or orthogonality. As more mixed strategy scenarios are explored, the understanding of the subsystems and the interrelationships among them can evolve. If designers change the model formulation and post-solution analysis, the solution space can be explored sufficiently, but in the original paper (Smith et al. 2015), the authors do not consider this. Therefore, we address the exploration of subsystems and reorganize them to improve the system performance of energy efficiency as well as to obtain knowledge on the nature of the system.

With the ALWC method, knowledge about the goals can be explored, captured, and reused in other problems. Here, we assume there is no knowledge of interrelationships of the goals or subsystems, or that the intuitive divisions of the system (four components—the turbine, condenser, pump, and heat exchanger) may be wrong. We use the ALWC method to learn about the subsystems and organization.

Table 2 The clustering results from the iterations

Interrelationship	Iteration 1: 1-level			Iteration 2	Iteration 3
Angle-based correlation	6 weight vectors	21 weight vectors	71 weight vectors	22 design scenarios by alternating levels {3, [[1, 6], [2, 4], [3, 5]]}	22 design scenarios by alternating levels {3, [[1], [2, 4], [3, 5, 6]]}
	{3, [[1], [2, 4], [3, 5, 6]]}	{3, [[1, 6], [2, 4], [3, 5]]}	{3, [[1, 6], [2, 4], [3, 5]]} (returned)	{3, [[1], [2, 4], [3, 5, 6]]}	{3, [[1], [2, 4], [3, 5, 6]]} (converged)
Orthogonality	6 weight vectors		21 weight vectors	22 design scenarios by alternating levels {3, [[1, 2, 4], [3, 5], [6]]}	22 design scenarios by alternating levels {3, [[1, 2, 4], [3, 5], [6]]}
	{3, [[1, 2, 4], [3, 5], [6]]}		{3, [[1, 2, 4], [3, 5], [6]]}	{3, [[1, 2, 4], [3, 5], [6]]}	{3, [[1, 2, 4], [3, 5], [6]]} (converged)

4 Results and discussion

4.1 Clustering result

By running the ALWC loop and applying different interrelationship methods and clustering methods, we obtain clustering results iteratively and list them in Table 2. For each interrelationship method, although only one clustering result converged, while running the ALWC loop, different clustering results are found. There are three clustering results which can be used to update leveling. They are summarized in Table 3. We assign some tentative meanings of the clusters attempting to interpret the cluster results intuitively. The intuitive interpretations may or may not be correct, so we shall not completely rely on them to evaluate and select one cluster result above the others. Our aims in processing the ALWC loop include identifying the various cluster results, obtaining more solutions that better complete the goals or improve the diversity of the tradeoff scenarios among the completion of goals, and discovering knowledge for (interrelationship measurement) method selection. Therefore, we use all cluster results and solutions in all iterations to enlarge the solution pool for decision support.

4.2 Improvement in goal achievement during the design scenario expansion

For a multi-goal problem, because the tradeoff between two goals often affects other goals, there are many nondominated and weakly dominated solutions; therefore, it is ineffective to use "solution domination" to rank the solutions. We use statistics to determine whether the results have been improved and enriched by iterating. In Table 4, we show the mean, standard deviation, best (minimum), and worst (maximum) deviations of each of the six goals, for each clustering scenario, and highlight the best case among all clustering scenarios. To ensure the results from different clustering scenarios are comparable, we use the actual deviation values instead of normalized values. For the 1-level scenario, we use the results of the 71 weight vectors to obtain the statistics, because these 71 weight vectors include the 6 and 21 weight vectors in previous iterations of the inner loop. For all the other three clustering scenarios, as each of them has 22 design scenarios, we use them to calculate statistics. For example, for Goal 1, among the means of each of the four clustering scenarios, the best (smallest) value 0.04 occurs in the third clustering scenario. The numbers in Table 4 have been rounded, but the longer computed numbers have been used for the comparison and the best ones are highlighted.

By leveling the goals using three clustering scenarios, the mean, standard deviation, and the worst-case results of the deviations of five goals excluding Goal 1 are improved. The worst case of the sum of all goals is improved. These observations indicate that by clustering and leveling the goals using the ALWC method, we identify better design scenarios for reducing the deviation (or improving the achievement) of most goals. In Fig. 9, we give box plots of the six goals under the Pre-emptive strategies (Fig. 9a), Archimedean or weighted sum strategy (Fig. 9b), and the ALWC (Fig. 9c). The vertical axes are the values of deviation variables. The six boxes in each graph are the box plots of the six deviations of all the scenarios used in each strategy. Shorter boxes close to the horizontal axes are preferred. We conclude that using ALWC, the deviations of Goals 1, 4, and 6 are substantially decreased.

Other statistics can be used to evaluate the iterative results from different perspectives. Designers may select or develop customized statistics based on the characteristics of specific problems.

4.3 Reducing the Euclidean distance to the Utopia point

For a multi-goal design problem, goals may represent the performance of various subsystems of the design. It is possible that the improvement of one goal results in a greater

Table 3 The summary of the cluster results returned to update the leveling

	Cluster result	Tentative meaning
1	{3, [[1], [2, 4], [3, 5, 6]]}	Cluster 1: Turbine moisture
		Cluster 2: Rankine cycle efficiency
		Cluster 3: Temperature exchanger/heat transfer efficiency
2	{3, [[1, 6], [2, 4], [3, 5]]}	Cluster 1: Turbine moisture and heat transfer effectiveness
		Cluster 2: Rankine cycle efficiency
		Cluster 3: Temperature exchanger efficiency
3	{3, [[1, 2, 4], [3, 5], [6]]}	Cluster 1: Turbine moisture and Rankine cycle efficiency
		Cluster 2: Temperature exchanger efficiency
		Cluster 3: Heat transfer effectiveness

Table 4 Statistics of the results

Statistics	Clustering scenario	$d_k = \max\{d_k^-, d_k^+\}$						Sum $\sum_{k=1}^{6} d_k$
		d_1	d_2	d_3	d_4	d_5	d_6	
Mean	1 level	0.05	0.81	0.36	0.81	0.90	0.02	2.94
	{3, [[1], [2, 4], [3, 5, 6]]}	0.05	0.78	0.51	0.78	0.92	0.01	3.04
	{3, [[1, 6], [2, 4], [3, 5]]}	0.04	0.84	0.34	0.84	0.90	0.01	2.96
	{3, [[1, 2, 4], [3, 5], [6]]}	0.05	0.80	0.45	0.80	0.91	0.01	3.01
Standard deviation	1 level	0.05	0.08	0.41	0.08	0.05	0.04	0.31
	{3, [[1], [2, 4], [3, 5, 6]]}	0.05	0.08	0.38	0.08	0.05	0.01	0.26
	{3, [[1, 6], [2, 4], [3, 5]]}	0.05	0.05	0.24	0.05	0.04	0.01	0.19
	{3, [[1, 2, 4], [3, 5], [6]]}	0.04	0.08	0.37	0.08	0.05	0.01	0.25
Minimum (best case)	1 level	0.00	0.69	0.04	0.69	0.84	0.00	2.67
	{3, [[1], [2, 4], [3, 5, 6]]}	0.00	0.69	0.06	0.69	0.85	0.00	2.72
	{3, [[1, 6], [2, 4], [3, 5]]}	0.00	0.69	0.06	0.69	0.85	0.00	2.72
	{3, [[1, 2, 4], [3, 5], [6]]}	0.00	0.69	0.07	0.69	0.85	0.00	2.73
Maximum (worst case)	1 level	0.12	0.88	0.90	0.88	0.97	0.16	3.52
	{3, [[1], [2, 4], [3, 5, 6]]}	0.12	0.87	0.90	0.87	0.97	0.02	3.31
	{3, [[1, 6], [2, 4], [3, 5]]}	0.12	0.87	0.90	0.87	0.97	0.02	3.29
	{3, [[1, 2, 4], [3, 5], [6]]}	0.12	0.87	0.90	0.87	0.97	0.02	3.29

loss in another goal or several other goals; see Fig. 10, as Goal 3 is improved 20% at D_2^{123} versus D_1^{123}, Goal 1 and 2 are worse by 80%. This may be desired in some situations, but more often than not, designers would rather avoid this, because overall performance may not be enhanced practically by improving a single subsystem. Therefore, we use the Euclidean distance to the Utopia point of the deviations, to evaluate the comprehensive performance of the achievement of all goals. Equation is the Euclidean distance to the Utopia point of the result from Design Scenario a for a K-goal problem. We statistically evaluate the evolution of the Euclidean distance and summarize it in Table 5. Using the clustering scenarios obtained during iteration, the mean, standard deviation, and the worst case of the Euclidean distance to the Utopia point are improved, but the best case is not improved. All the design scenarios and corresponding results from all iterations are added to the solution pool for designers to select. Designers may customize their post-solution analyses for further decision support on design scenario generation and selection

$$|OD_a| = \left(\sum_{k=1}^{K} {d_{ak}}^2 \right)^{\frac{1}{2}}. \tag{9}$$

4.4 Reducing computational complexity

Smith and coauthors explore the leveling of all six goals and simplify the scenarios using their expertise. The theoretical number leveling scenario is 6! = 720, but they select 15 scenarios that they think are most representative (Smith et al. 2015). If designers use weight vectors to explore the combination of the goals, the number of weight vectors is based on how many pieces they divide the goal weight in the range of [0, 1]; see Eq. 11

$$H = \binom{6+p-1}{p}, \text{for} p = 2,3,4,5,6\ldots, \quad H = 21,56,126,252,462\ldots. \tag{10}$$

For a K-goal problem, we explore combinations of the goals (design scenarios) using a mixture of Pre-emptive and Archimedean schemes and enumerate all possible design scenarios. These scenarios are necessary due to the lack of domain knowledge about the interrelationships among the goals or due to missing specific design preferences about tradeoffs among the goals. We need to explore Λ design scenarios where Λ in the range defined in Eq. 12, κ is the number of levels of the K goals, $\frac{K}{\kappa}$ is the average number of goals at each level on average (if it is not be an integer so it is rounded down), and p is the number of pieces that each goal's weight is divided into when the weights are assigned. Hence, the computational complexity of enumerating the mixture of Pre-emptive and Archimedean design scenarios is shown in Eq. 13

$$\sum_{\kappa=1}^{K} \kappa! \cdot \binom{\lfloor \frac{K}{\kappa} \rfloor + p - 1}{p} \leq \Lambda \leq \sum_{\kappa=1}^{K} \kappa! \cdot \binom{K+p-1}{p}, \tag{11}$$

$$O\left(K! \cdot K^{p-1}\right). \tag{12}$$

Using the ALWC, the goals are leveled based on clustering results and a specific number of levels are obtained. Using these levels, we do not need to enumerate all scenarios. For each clustering scenario, the number of leveling scenarios is reduced to $3! = 6$. Within each level, the goals are combined using weight vectors. The stopping criteria of weight vectors generation (Line 5 of Algorithm 2) help to prevent designers from using many unnecessary weight vectors. Using the angle-based correlation method to calculate an interrelationship matrix, we explore 142 design scenarios in three iterations and converge. Using the orthogonality method, we then explore 71 design scenarios in three iterations and converge.

For a K-goal problem, if we explore design scenarios using the ALWC, a number of clusters, κ, can be identified, based on the goals' interrelationships, and on average, there are $\frac{K}{\kappa}$ goals in each cluster and we need Λ' design scenarios; see the range of Λ' in Eq. 14. The computational complexity of using ALWC is shown in Eq. 15. When $> \kappa$ and $\kappa > 1$, Λ' is smaller than Λ (Eq. 15), we do not need to exhaustively study the cases when κ is equal to all integers between 1 and K

$$\kappa! \cdot \binom{\lfloor \frac{K}{\kappa} \rfloor + p - 1}{p} \leq \Lambda' \leq \kappa! \cdot \binom{\lceil \frac{K}{\kappa} \rceil + p - 1}{p}, \tag{13}$$

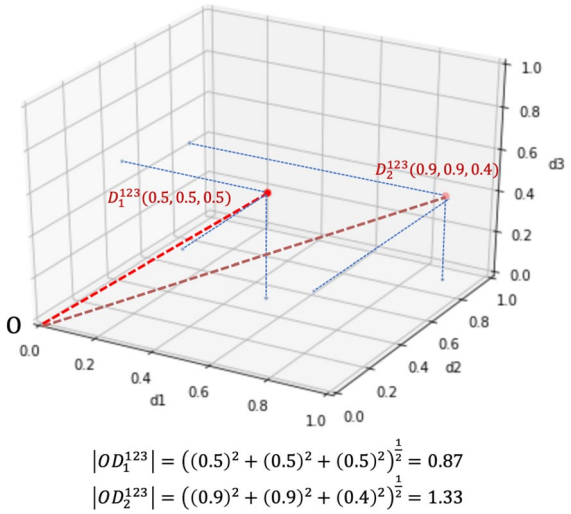


Fig. 10 An example of improving Goal 3 by 20% while worsening Goal 1 and Goal 2 by 80%, respectively

Table 5 Statistics of the Euclidean distance to the Utopia point of the results under each clustering scenario. The Euclidian distance is the length of the line segment between the two points.

	Mean	Standard deviation	Minimum (best case)	Maximum (worst case)
1 level	1.56	0.08	*1.45*	1.70
{3, [[1], [2, 4], [3, 5, 6]]}	1.57	0.08	1.46	1.66
{3, [[1, 6], [2, 4], [3, 5]]}	*1.54*	*0.06*	1.46	*1.64*
{3, [[1, 2, 4], [3, 5], [6]]}	1.56	0.07	1.46	*1.64*

$$O\left(\kappa! \cdot \left(\frac{K}{\kappa}\right)^{p-1}\right), \tag{14}$$

$$\Lambda' < \Lambda, \text{if} K > \kappa \text{and} \kappa > 1. \tag{15}$$

4.5 Verification of the results

4.5.1 The results are verified using domain knowledge

In all three clustering scenarios, Goals 2 and 4 are always in one cluster, and Goals 3 and 5 are always in one cluster, whereas the clustering result of Goal 1 and Goal 6 change in various iterations. This implies that Goals 2 and 4 are strongly correlated or weakly orthogonal, as are Goals 3 and 5; the relationships between Goals 1 and 6 and the other goals are not significant. The clusters represent the subsystems, which verify the clustering result; see Table 6.

Table 6 Meaning of the three clusters

Meaning	Representative goals
Rankine cycle efficiency	Goal 2: Maximize Rankine cycle efficiency
	Goal 4: Maximize system efficiency indicator 1
Heat exchange efficiency	Goal 3: Maximize temperature exchanger efficiency
	Goal 5: Maximize system efficiency indicator 2
Moisture in turbine	Goal 1: Moisture in steam leaving the turbine
Heat transfer effectiveness in exchanger	Goal 6: Heat transfer effectiveness in exchanger

Goals 2 and 4 are related to the efficiency of the Rankine cycle and the system efficiency indicator 1 is to increase the Rankine cycle efficiency. Goals 3 and 5 represent heat exchange efficiency. The temperature exchanger and heat transfer work are synergetic in the working system. Goal 1 is about the moisture in the turbine, and it sometimes forms a single cluster and sometimes is clustered with Goals 2 and 4. Our domain knowledge confirms that when the moisture in the turbine is low, the Rankine cycle is very efficient, but the turbine is a relatively isolated subsystem. Goal 6 is sometimes a single cluster and sometimes clustered with Goals 3 and 5. These goals all deal with the efficiency of the heat exchanger; Goals 3 and 5 are more about the temperature, whereas Goal 6 deals with the liquid flow. However, in one clustering scenario, Goals 1 and 6 are in one cluster. The scatter plots of any two goals help us to interpret this phenomenon, Fig. 11. The deviations of Goal 1 and Goal 6 are relatively small and close to the Utopia point O (Fig. 11e) in comparison with the other two-goal plots. Even after normalizing the deviations in the range [0, 1], Goals 1 and 6 are weakly correlated. However, when such a clustering scenario was used in the next iteration, the clustering results "returned to normal." This indicates that iteratively clustering and updating is necessary, because it can expand the sample size and remove bias.

Using domain knowledge, the clustering result has been verified for this thermal system design problem. For multi-goal problems in which domain knowledge is missing, the ALWC method can assist designers in identifying the interrelationships among the goals. With the ALWC, more design scenarios regarding the combination forms of goals are explored based on their interrelationships. These design scenarios and deviations of the goals are added to the solution pool for designers to select among to satisfy different requirements, improve the problem formulation, and enable further analysis to obtain a better understanding of the interrelationship among subsystems. The ALWC method can be used as a tool to partition a design problem, especially when information is incomplete.

5 Closing remarks

In this paper, we develop a domain-independent method to capture and reuse the knowledge of a multi-goal engineering-design problem to facilitate the exploration and selection of design scenarios. To accomplish this, the Adaptive Leveling-Weighting-Clustering (ALWC) for exploring multiple design scenarios is proposed. We use a thermal system test problem to illustrate the effectiveness of the ALWC method.

Using the ALWC method, with increasing weight vectors, the interrelationships among goals based on their deviations, or achievement rates evolve and converge. Based on their interrelationships, goals are grouped into clusters to represent different subsystems. The combinations of the goals are explored iteratively, using either the Pre-emptive or Archimedean strategy. This facilitates the assignment of each cluster a different level (leveling) and a combination of the goals in each level using weight vectors (weighting). Through iteration, more design scenarios are identified, and corresponding solutions are obtained for designers to choose the appropriate design scenario and then improve the design. As a tool to acquire insight when domain knowledge is lacking, the combination of the goals is explored, so that better solutions regarding the average deviations, standard deviations, worst case, and the Euclidean distance to the Utopia point are identified and computational complexity is reduced.

The algorithms embodied in the ALWC method can be extended, modified, or customized for specific requirements. When there is insufficient expertise to support decision-making, subsystem division and tradeoffs, and design improvement, the knowledge discovered using the ALWC method can be used to explore the ways that may contribute to design improvement. Our assumption in using the ALWC algorithm is that the domain expertise on the interrelationships among the goals is missing and the solutions can accurately reflect their interrelationships. If domain expertise is sufficient to enable decision-makers to determine subjective preferences, the ALWC may not deliver a better solution regarding computational complexity or satisfying multiple requirements.

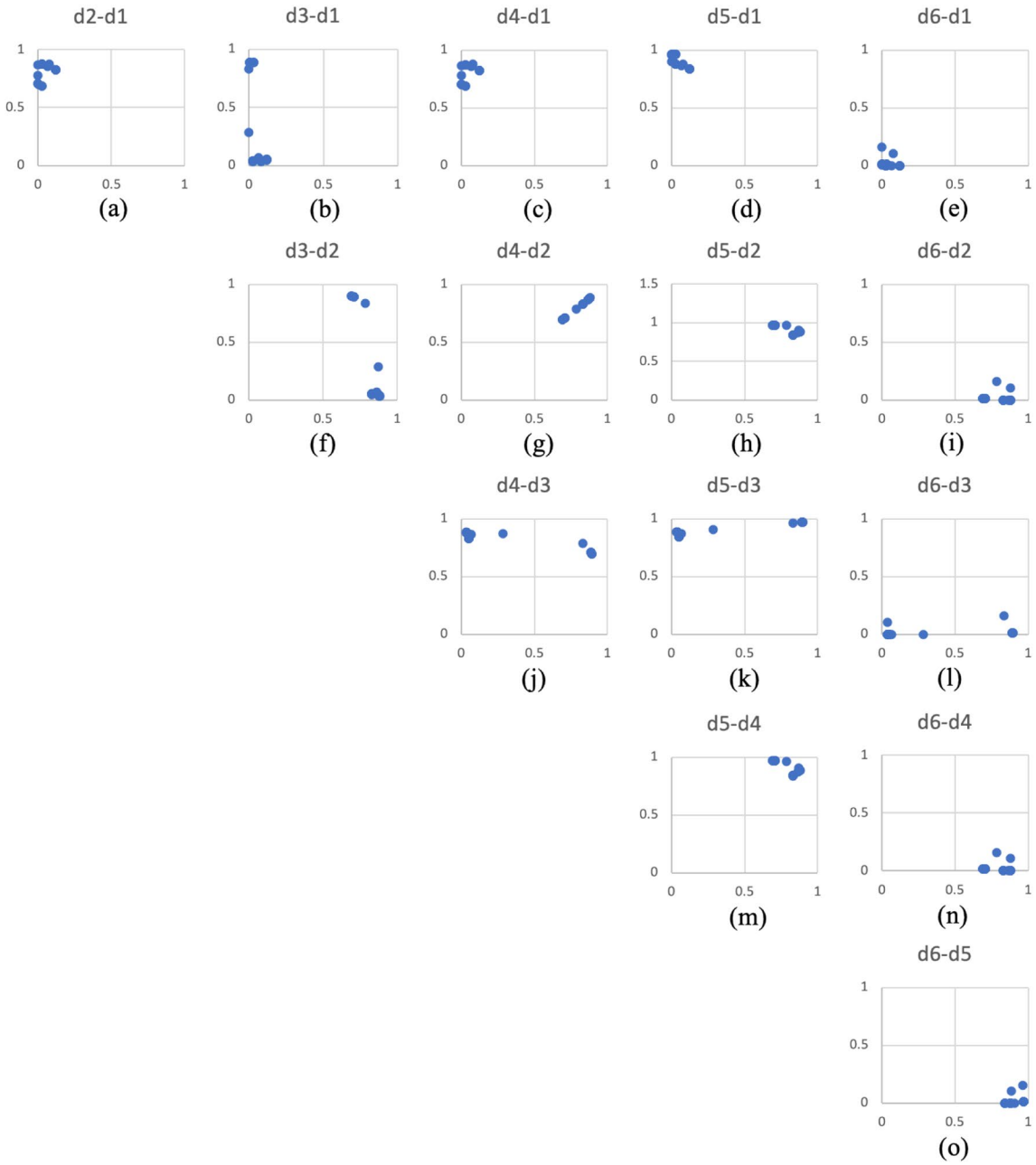


Fig. 11 Scatter plots of any two goals using deviations of 1-level, 21 weight vectors—the vertical and horizontal axes of each sub-figure represent the deviation of two goals, respectively. For example, in Fig. 12a, the vertical axis is d_2 and the horizontal axis is d_1

Appendix: The Algorithms in the ALWC

There are six steps in the ALWC. The model is formulated as a compromise Decision Support Problem (cDSP) (Mistree et al. 1981, Mistree, Hughes et al. 1993) (see definition in Glossary) in Step 1.1. The number of clusters, C, is initialized to "1" in Step 1.2.[4]

In Step 2 Leveling, starting from the second iteration, the goals are leveled based on clustering results—the output of Step 5 in the previous iteration. In Step 1, the number of levels, n, is updated. In Step 2.2, each cluster in turn is set to Level 1 to Level n. The algorithm is as follows.

See Algorithm 1[5] and 2[6].

Algorithm 1 Step 2 Leveling

2.1 Updating

1: **Given** Cluster results $\{C, Cluster\}$, cDSP, Leveling information $\{L, Level\}$ // C, L are integers representing the number of the clusters and the number of levels respectively. "$Cluster, Level$" are two-dimension arrays containing the clustering result[6] and leveling information.
2: integer $L = C$
3: array $Level = Cluster$
4: **return** $\{L, Level\}$ and **go to** Step 3 Weighting

2.2 Alternating levels

5: **if** $L \geq 2$
6: array temp $= Level[L]$
7: **for** integer n in range $[2, L]$
8: $Level[n] = Level[n-2]$
9: $Level[1] = temp$
10: **return** $\{L, Level\}$ and **go to** Step 3 Weighting

[4] This footnote item supports the "clustering result" in Algorithm 1, 2.1 Updating, Line 2., E.g.,

Cluster [[*G1, G2, G 4*], [*G3, G5, G6*]] means that there are two clusters: Cluster 1 includes Goal 1, Goal 2, and Goal 4, whereas Cluster 2 includes Goal 3, Goal 5, and Goal 6. Cluster [2] [3] represents the third goal in Cluster 2, and so, *Cluster* [2][3]=*G6*. The array *Level* works in the same way.

[5] XXX.

[6] E.g., $\text{Weight} = \left[\left[[1,0],[0,1],\left[\frac{1}{2},\frac{1}{2}\right]\right],[[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]]\right]$ means that there are two levels; Level 1 has two goals and there are three vectors for the two goals; Level 2 has four goals and there are four weight vectors for the four goals. *Weight*[*n*] represents the weight vectors of the goals in the *n*th level; *Weight*[*n*][*m*] represents the *m*th weight vector of the goals in the *n*th level. *Weight*[*n*][*m*][*k*] represents the weight of the *k*th goal of the *m*th weight vector in the *n*th level. In this example, $Weight[1][3][1] = 1/2$. This footnote item is related to Algorithm 2, Line 1.

Algorithm 2 Procedure 3 Weighting

1: **Given** $\{L,\ Level\}$, $\{C,\ Cluster\}_{i\text{-}1}$, *Weight* // "*Weight*" is a three-dimension array containing weight vectors[7]

2: **Initiate parameters:**
integer $p = 2$, $i = 1$
array $F = [\ ]$, $wv = [\ ]$, $Weight = [wv]$, $\sigma(\mathcal{D}_0) = 0,\ \varepsilon = \mathcal{E}$

3: **for** integer n in range $[1, L]$

4: integer $M = length\ (Level[\ n\ ])$ // Assign the number of goals in Level n as M

5: **while** $|\ \sigma\ (\mathcal{D}_i) - \sigma(\mathcal{D}_{i-1})\ | > \varepsilon$

6: $i = i + 1$

7: **for** integer q in range $[1, p]$

8: **for** integer r in range $[\ 0, q\ ]$

9: $F = F \cup \frac{r}{q}$

10: **for** I in $product\ (* [F]\ * M)$ // "I" is a one-dimensional array with three elements that are the results of the cross product of M and F

11: **if** $sum(I) ==$ 1// If the sum of the elements in "I" equals one.

12: $wv = I$

13: $Weight\ [\ n\ ] = Weight\ [\ n\ ] \cup wv$

14: **Call** Procedure 4 and **return** $\mathcal{D}_i$

15: **Calculate** $\sigma(\mathcal{D}_i)$

Glossary

ALP — Adaptive linear programming algorithm. A second-order sequential linear programming algorithm. Using the ALP to solve a cDSP, one can obtain *satisficing* solutions that are relatively insensitive to certain uncertainties and complexities; see (Guo 2021) for details on why and how the cDSP and ALP can return *satisficing* solutions

Archimedean — A strategy of managing multiple goals by compromising the achievement of the goals. Also known as a weighted sum method (Ignizio 1976, Guéret et al. 1999)

cDSP — Compromise Decision Support Problem. A model formulation construct combining mathematical programming and goal programming. Using a cDSP, designers minimize the deviation between the achieved value of a goal and its target value. The ALP is used to solve a cDSP to obtain *satisficing* solutions

Deviation — Deviations are measured both above and below the target values of goals. These unwanted deviations are then minimized in an achievement function

Diverse Solutions

Goal — In this paper, the term "goal" refers to the objective of a design problem when its target value is known. A problem is solved by minimizing the deviation between the achieved values and the goal target. Therefore, a problem is solved by maximizing the goal achievement (Guo 2021).[3]

Goal achievement rate — The rate at which the target value of a goal is achieved

Goal level — Or level of goals, when applying Pre-emptive strategy to manage multiple goals, the goals that are placed in the same priority to be dealt with are in one level, so the goal level indicates the priority of the goals in that level

Multi-goal problems — Problems with more than one goal

[3] The reason that we use goals instead of objectives for managing engineering-design problems is that using goals, we obtain solutions that meet the necessary KKT conditions. Using objectives, it is necessary to meet both the necessary and sufficient KKT conditions, which reduces the chance of identifying a solution for a nonlinear, nonconvex problem. The detailed reasoning is in Sect. 2.2.7 of Guo (2021).

Pre-emptive	A strategy of managing multiple goals by decentralizing the problem. Also known as a lexicographic approach (Kortanek and Maxwell 1969). The goals are placed in multiple levels of priority. The first level goal function is satisfied as far as possible and then it is held within a tolerance and the second level goal function is addressed, and so on in an attempt to address all the goals across all levels (Ignizio 1976)
Robustness	The ability of a system to be insensitive to variations or uncertainties
Satisficing	A decision-making strategy that entails searching through available alternatives until an acceptability threshold is met (Byron 1998)
Satisficing solutions	Solutions that are not necessarily optimal but adequate. These solutions are obtained by minimizing the distance between what the system can achieve and the ideal case (Simon 1996) using the ALP (Mistree 1993)
Scalarization	Reducing multiple goals to a single function that can be solved as a single goal (Bandrau et al. 2017). Two of the most common scalarization methods are Pre-emptive ordering (Lexicographic) and weighted sums (Archimedean)

Acknowledgements Lin Guo acknowledges Shehnaz Shaik for programming contributions. Lin Guo acknowledges the financial support from the Pietz Professorship and Start-Up Fund at the South Dakota School of Mines and Technology. Lin Guo and Janet K. Allen gratefully acknowledge the financial support from the John and Mary Moore Chair at the University of Oklahoma. Lin Guo and Farrokh Mistree gratefully acknowledge financial support from the L.A. Comp Chair at the University of Oklahoma. Ru Wang gratefully acknowledges the Project funded by China Postdoctoral Science Foundation [Grant 2018M640073]. Financial support was provided from the Office of the Vice President for Research and Partnerships and the Office of the Provost, University of Oklahoma for assistance with publication charges.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Replication of results The data and codes for this paper will be submitted as supplementary material.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Avigad G, Matalon EE (2011) The multi-single-objective problem and its solution by way of evolutionary algorithms. Res Eng Design 22(2):87–102

Avigad G, Moshaiov A (2009) Interactive evolutionary multiobjective search and optimization of set-based concepts. IEEE Trans Syst Man Cybern Part B (cybernetics) 39(4):1013–1027

Bader J, Zitzler E (2011) HypE: an algorithm for fast hypervolume-based many-objective optimization. Evol Comput 19(1):45–76

Byron M (1998) Satisficing and optimality. Ethics 109(1):67–93

Chen W, Allen JK, Mistree F (1997) A robust concept exploration method for enhancing productivity in concurrent systems design. Concurr Eng 5(3):203–217

Chiu M-C, Chung W-H, Lin H-H (2021) Applying DFA and goal programming to improve economic efficiency, material handling convenience, and sustainability of a product packaging system. Res Eng Design 32(2):157–173

Choi H-J, Austin R, Allen JK, McDowell DL, Mistree F, Benson DJ (2005) An approach for robust design of reactive power metal mixtures based on non-deterministic micro-scale shock simulation. J Comput Aided Mater Des 12(1):57–85

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Fishburn PC (1974) Exceptional paper—lexicographic orders, utilities and decision rules: a survey. Manage Sci 20(11):1442–1471

Guéret C, Prins C and Sevaux M (1999) Applications of optimization with xpress-MP. contract: 00034

Guo L (2021) Model evolution for the realization of complex systems. Ph.D., University of Oklahoma

Hao F, Merlet J-P (2005) Multi-criteria optimal design of parallel manipulators based on interval analysis. Mech Mach Theory 40(2):157–171

Ignizio JP (1976) An approach to the capital budgeting problem with multiple objectives. Eng Econ 21(4):259–272

Jaimes AL, Coello CAC, Barrientos JEU (2009) Online objective reduction to deal with many-objective problems. International conference on evolutionary multi-criterion optimization. Springer

Katsikopoulos KV (2011) Psychological heuristics for making inferences: definition, performance, and the emerging theory and practice. Decis Anal 8(1):10–29

Kortanek K, Maxwell W (1969) On a new class of combinatoric optimizers for multi-product single-machine scheduling. Manage Sci 15(5):239–248

Lemmon EW, Huber ML (2013) Implementation of pure fluid and natural gas standards: reference fluid thermodynamic and transport properties database (REFPROP). National Institute of Standards and Technology, NIST

Macquorn Rankine WJ (1853) XVIII. On the general law of the transformation of energy. Lond Edinburgh Dublin Philos Mag J Sci 5(30):106–117

Meier C, Yassine AA, Browning TR, Walter U (2016) Optimizing time–cost trade-offs in product development projects with a multi-objective evolutionary algorithm. Res Eng Design 27(4):347–366

Mistree F, Hughes OF, Phuoc HB (1981) An optimization method for the design of large, highly constrained complex systems. Eng Optim 5(3):179–197

Mistree F, Hughes OF, Bras B (1993) The compromise decision support problem and the adaptive linear programming algorithm. In: Kamat MP (ed) Structural optimization: status and promise. AIAA, Washington

Mistree F, Patel B, Vadde S (1994) On modeling multiple objectives and multi-level decisions in concurrent design. Adv Design Autom 69(2):151–161

Mistree F, Hughes, OF, Bras B (1993) The compromise decision support problem and the adaptive linear programming algorithm. Structural Optimization: Status and Promise, (Ed. M.P. Kamat), AIAA, Washington DC, 247-286.

Moon SK, Park KJ, Simpson TW (2014) Platform design variable identification for a product family using multi-objective particle swarm optimization. Res Eng Design 25(2):95–108

Previati G, Ballo F, Gobbi M (2019) Concurrent topological optimization of two bodies sharing design space: problem formulation and numerical solution. Struct Multidiscip Optim 59(3):745–757

Reynoso-Meza G, Blasco X, Sanchis J, Herrero JM (2013) Comparison of design concepts in multi-criteria decision-making using level diagrams. Inf Sci 221:124–141

Sabeghi M, Shukla R, Allen JK and Mistree F (2015) Solution space exploration of the process design for continuous casting of steel. In: ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers

Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985, Carnegie-Mellon University, Pittsburgh, PA, Lawrence Erlbaum Associates. Inc., Publishers

Seada H and Deb K (2014) U-NSGA-III: a unified evolutionary algorithm for single, multiple, and many-objective optimization. COIN Report 2014022.

Simon HA (1996) The sciences of the artificial. MIT Press

Soltani AR, Tawfik H, Goulermas JY, Fernando T (2002) Path planning in construction sites: performance evaluation of the Dijkstra, A∗, and GA search Algorithms. Adv Eng Inform 16(4):291–303

Tang D, Zhu R, Tang J, Xu R, He R (2010) Product design knowledge management based on design structure matrix. Adv Eng Inform 24(2):159–166

Toffler, A. (1970). Future shock, Bantam.

Wang F-Y (1994) On the extremal fundamental frequencies of one-link flexible manipulators. Int J Robot Res 13(2):162–170

Wang R, Nellippallil AB, Wang G, Yan Y, Allen JK, Mistree F (2018) Systematic design space exploration using a template-based ontological method. Adv Eng Inform 36:163–177

Wikipedia (2019) Rankine cycle. from https://en.wikipedia.org/wiki/Rankine_cycle

You Z-H, Smith S (2016) A multi-objective modular design method for creating highly distinct independent modules. Res Eng Design 27(2):179–191

Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731

Zitzler E, Laumanns M and Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. TIK-Report 103

Mistree F, Hughes OF, Bras B (1993) the compormise decision support problem and theadaptive linear programming algorithm, Structural Optimizatiion Status and Promise, (Ed. M.P. Kamat), AIAA,Washington, DC, 247-286.

https://dspace.org/

Managing multi-goal design problems using adaptive leveling-weighting-clustering algorithm

Guo, Lin; Milisavljevic-Syed, Jelena; Wang, Ru; Huang, Yu; Allen, Janet K. ...

2022-09-25
Springer
https://doi.org/10.1007/s00163-022-00394-z
Guo L, Milisavljevic-Syed J, Wang R, et al., (2023) Managing multi-goal design problems using adaptive leveling-weighting-clustering algorithm, Research in Engineering Design, Volume 34, Issue 1, January 2023, pp. 39–60

Downloaded from a DSpace repository. https://dspace.org/