

Research Article

A Design-Task-Oriented Model Assignment Method in Model-Based System Engineering

Xiaofei Wang ^{1,2}, Wenhe Liao,¹ Yu Guo,¹ Daoyuan Liu,¹ and Weiwei Qian¹

¹College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

²Nanjing Research Institute of Electronics Technology, Nanjing 210039, China

Correspondence should be addressed to Xiaofei Wang; wangxiaofei@nuaa.edu.cn

Received 4 May 2020; Revised 19 July 2020; Accepted 6 August 2020; Published 24 August 2020

Guest Editor: Juan Carlos Leyva-Lopez

Copyright © 2020 Xiaofei Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In model-based system engineering (MBSE), reuse of existing models in the development of a new system can be advantageous. Automatic assignment of existing models to each design task within a design task set has been proven to be feasible. However, while several studies have discussed the significance of models in MBSE and methodologies for models reuse, solving the model reusability problem through a model assignment method has not been discussed. Additionally, a significant challenge in model assignment is to address the conflict between the maximization of the model value summations, which are yielded by assigning the models to a design task set, and the minimization of the execution cycle of the task set. This study (a) proposes a design-task-oriented model assignment method that establishes a multiobjective model, based on a model assignment integration framework, and (b) designs a differential-evolution-combined adaptive nondominated sorting genetic algorithm-II to provide an optimal tradeoff between maximizing the total model values and minimizing the execution cycle of the task set. By comparing the performance of the algorithm in resolving the assignment of models to a design task set with those of two conventional algorithms in a phased-array radar development project, the algorithm's performance and promotion of system development are verified to be superior. The new method can be applied for developing model scheduling software for MBSE-compliant product development projects to improve using effects of the models and development cycle.

1. Introduction

Considering the increasingly sophisticated customer demands and the growing requirements for increased product development capabilities—given that products are more integrated and intelligent in various industries, including aviation and space—the traditional product development mode is no longer satisfactory [1]. Model-based system engineering (MBSE) accomplishes the development of complex products with a new mode and is capable of forecasting product behaviors, thereby improving the productivity of the product development process. Arguably, the developments of aviation and space systems, which are regarded as the most complex cyber-physical systems (CPSs) in the industry [2], have adopted the MBSE approach to facilitate the implementation of all the phases of the product lifecycle [3–8]. The widespread use of MBSE has tended to

shift emphasis from data management to model management throughout the entire product's lifecycle [9].

MBSE is the formalized application of modeling used to support system requirements, design, analysis, and verification and validation activities, beginning with the conceptual design phase and continuing throughout the development and later lifecycle phases [10]. The output of the MBSE activities is a coherent model of the system (i.e., system model), whereby the emphasis is placed on the evolution and the refinement of the model using model-based methods and tools [11]. Therefore, the model plays the most important role in each stage of the product's lifecycle.

In the concept and design phases, a shared system model is needed to support the exchange of information across various aspects. Accordingly, this system model serves as the core model of the system to provide information and maintain consistency with domain-specific models. For

example, the Space Systems Working Group of International Council on Systems Engineering developed the CubeSat reference model for mission-specific CubeSat teams [6]. The German Aerospace Center presented the conceptual data model as an abstraction of domain-specific models [5]. In the development of mechatronic systems, Barbieri et al. used the conceptual model as the information source for domain-specific models of every functional module [12]. Researchers from nine leading Chinese academic and industrial institutions have gathered to discuss the definition and application of MBSE-compliant product meta-models [9].

MBSE has also found application in the early tender phases of complex CPS development, wherein the complex CPS customer can use the MBSE approach to generate a model-based request for tenders and pass it on to the supplier who can use the model to perform system development. Australia's Defense Science and Technology Organization pioneered the adoption of a whole-of-system analytical framework for information transfer across the contractual interface [8]. The integrated design methodology proposed in [12] also demonstrated that the MBSE approach can be used to fulfill stakeholder requirements by adopting them in the system requirements. In addition to the above stages, MBSE can also be used as an effective method for product manufacturing system planning in the manufacturing phase [13, 14].

At a higher level, a model-based system analysis framework is needed to provide the capability to access, integrate, and transform disparate data into actionable information for the design and analysis of complex systems. The United States (U.S.) Air Force (AF) established the digital thread initiative [3, 15] that generates an engineering analytical framework based on an authoritative digital surrogate representation throughout the entire product's lifecycle.

The most commonly used modeling language in MBSE is the system modeling language (SysML), a general-purpose graphical modeling language that supports analysis, specification, design, verification, and validation of complex systems [11, 16], and has been adopted by many MBSE projects [1, 9, 12, 13, 17]. Despite the fact that it is accepted by the Object Management Organization as a standard modeling language, SysML is not easy to adapt for system engineers who have not been exposed to object-oriented concepts because, like the unified modeling language (UML), it emphasizes familiarity with these concepts. For ease of use, organizations have developed some modeling languages, including the modeling and analysis of real-time embedded (MARTE) systems [18], architecture analysis and design language (AADL) [2], domain-specific modeling language (DSML) [19], and others. Based on these languages, some powerful MBSE platforms and tools have been developed. For example, Thales' ARCADIA™ and Capella™ workbench [20, 21] support requirement analyses and system design in the areas of transportation, aviation, space, and radars, while Tucson Embedded Systems' AWESUM™ tool suite supports the U.S. Army's joint common architecture project [2]. In addition, the U.S. Department of

Defense's high-performance computing modernization program has developed a computational research engineering acquisition tools environment for air vehicles [3] and realized the digital thread of the U.S. AF.

With the increasing application of the MBSE approach in the industry, numerous models built using various modeling languages have been stored by various organizations adopting MBSE. However, the models are often not reused effectively. When faced with a new development project, the development teams often create new models, rather than reusing the existing models available within each discipline. This repetition of work amounts to an unnecessary expenditure of cost and time for the project. To reuse existing models, appropriate models should be identified and assigned to each of the design tasks in the development project. While literature on model reuse has described approaches for applying development environments [22], ontologies [23], and model repositories [24, 25], no previous studies have discussed model assignments. Therefore, this study focuses on the establishment of a design-task-oriented model assignment method to support model reuse in MBSE. The study includes the following main components.

- (i) An integration framework capable of assigning the models in the repository to the design task set is established. In existing MBSE approaches, simply integrating the tools into the product lifecycle development environment [2, 3] does not enable model reuse because it does not consider how the stored models are assigned to the design tasks of the project. The framework proposes an optimization scheme for matching the models and the design tasks to support the needs of the MBSE platform and tools for models reuse.
- (ii) The value of each model for a design task is quantified. This captures the suitability of the model for the task. The literature regarding model management in MBSE and software engineering has focused on model management platform [26], model repository building methods [27], and so on, but has not provided a method for the evaluation of model value for the task. This study applies an advantage-number-based analytical technique to evaluate the models to be assigned from the perspective of value and, accordingly, preferentially filters the models to establish a design-task-oriented preferred model set.
- (iii) A mathematics model for design-task-oriented model assignment and an optimization algorithm are proposed. This study suggests a multiobjective model of design-task-oriented model assignments to minimize the task set execution cycle and maximize the actual value summation of the models. Additionally, to solve the proposed multiobjective model, the study has designed a differential-evolution-combined adaptive nondominated sorting genetic algorithm-II (DA-NSGA-II).

Finally, based on a case study, the new algorithm is proven to have better performance and promotion of system development than the traditional non-dominated sorting genetic algorithm-II (NSGA-II) and particle swarm optimization (PSO).

The rest of this paper is organized as follows. The second section describes a design-task-oriented model assignment integration framework. In the third section, a multiobjective model of model assignment is established according to the quantification of model value. The fourth section proposes an improved NSGA-II to solve the multiobjective model. A case study based on a phased-array radar development project is discussed in the fifth section. Finally, the sixth section focuses on the study's conclusion and the potential avenues for future work.

2. Design-Task-Oriented Model Assignment Integration Framework

In the field of MBSE, model reuse is able to improve the efficiency and reduce the cost of system development. It is a feasible method to assign the existing models to the tasks within a design task set under the condition of discipline matching between the models and the tasks. Using the method, a suitable model is selected for each task according to the model value for a design task and the execution cycle of the task set following the assignment of the models to the task.

Owing to the different properties of the model for different tasks, such as integrality and reliability, the same model would yield different values when applied to various tasks. A model is considered to yield a high value if it can improve the execution effect of a task that it is assigned to. By contrast, the same model is considered to yield low value if it would worsen the execution effect of another task that it is assigned to. In general, model assignment attempts to achieve the highest possible summation of the models' actual values once the models are assigned to the tasks.

In addition, the execution cycle of a single task is different when different models are applied. For a given task set restricted by the temporal relation of the tasks, the applied model assignment strategy determines the execution cycle of the task set. Thus, to complete the task set as soon as possible, another objective of the model assignment is the minimization of the task set execution cycle.

For a simple system, the existing models can be manually assigned to each task in the design task set with ease. However, when the developed system is relatively complex, the model values for a task cannot be directly measured due to the complex model properties and wide task ranges. Thus, it is difficult to directly compare the values of a set of models for a particular task. Moreover, in the development of a complex system, a single model assignment scheme cannot simultaneously satisfy the requirements of maximizing the total model value and minimizing the task set execution cycle. The model assignment solutions must be optimized to determine the optimum scheme.

However, the optimization cannot be performed by a human due to the large number of models and tasks in a complex system.

Consequently, a framework for model assignment integration must be established in the field of MBSE in order to quantitatively evaluate the model values and perform an optimal tradeoff between the maximization of the total model values and the minimization of the execution cycle. By doing so, an optimum solution is obtained allowing the assignment of models to the design task set. The current study establishes a design-task-oriented model assignment integration framework, as presented in Figure 1. The key features of the framework are as follows.

- (i) A model repository needs to be defined, and the existing models of previous projects should be saved in the repository.
- (ii) The design task set is derived from task planning.
- (iii) The values of the models from the model repository, with respect to the design tasks, are evaluated quantitatively. Next, a preferred model set is generated after an optimal selection of models based on the quantitative values of the models.
- (iv) According to the preferred model set value and cycle matrices concerning the design task set, a multiobjective model that relates the maximization of the models' actual value summation and the minimization of the task set execution cycle is established.
- (v) The model assignment scheme is obtained by a multiobjective optimization algorithm.

3. Modeling of Design-Task-Oriented Multiobjective Model Assignment

3.1. Calculation of Model's Actual Value for Task. Based on the model assignment integration framework, models are selected from the model repository. This process involves the calculation of the actual values of different models for each of the specified tasks.

The given development project applies the mode of MBSE assuming that the task set $T = \{T_1, T_2, \dots, T_n\}$ has n models. Additionally, the model repository related to the task set has l models, with $M = \{M_1, M_2, \dots, M_l\}$ as the model set and $S = \{s_1, s_2, \dots, s_k\}$ as set of the models' attributes.

Definition 1. The value of the model M_i for task T_j is v_{ij} .

$$v_{ij} = \sum_{g=1}^k w_g c_j^i(s_g), \quad (1)$$

where $g = 1, 2, \dots, k$, $i = 1, 2, \dots, l$, $j = 1, 2, \dots, n$, w_g is the weight of the model's attribute, and $c_j^i(s_g)$, which is expressed by the five-demarcation method as $\{1, 3, 5, 7, 9\}$, is the value of the attribute s_g of model M_i after it is assigned to task T_j .

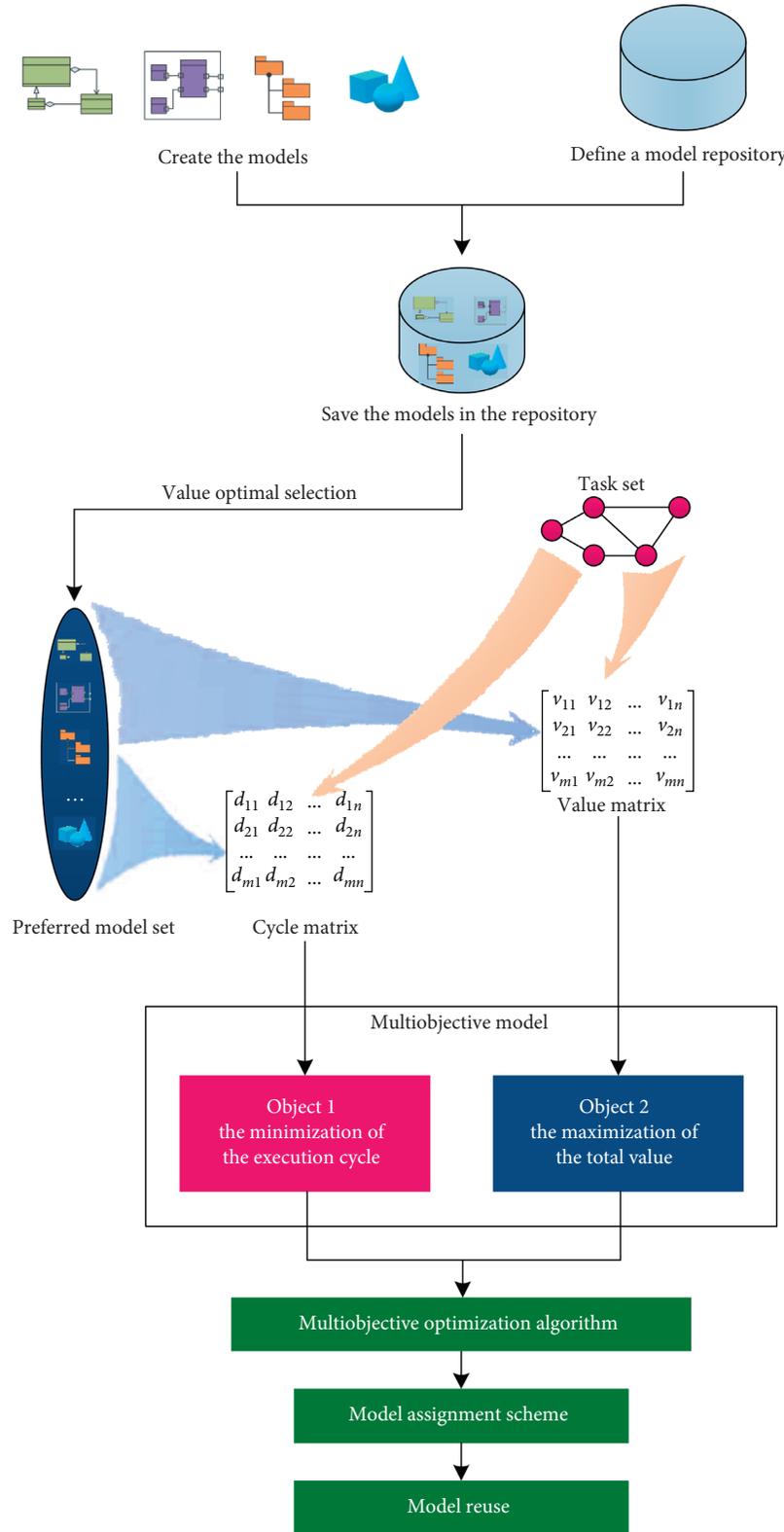


FIGURE 1: Design-task-oriented model assignment integration framework.

Here, the model's attribute weight w_g is determined using a method based on the significance of the Pawlak attribute in the decision table according to rough set theory [28]. To list the model values of the decision table, select the original assessment data of l' models from the model

repository as the universe of discourse U , condition attribute set S as the set of model attributes $\{s_1, s_2, \dots, s_k\}$, formulate the decision attribute set D as the set of model values $\{v\}$, and assign the field value according to $\{\text{high, middle, low}\}$. The decision table is thus obtained.

TABLE 1: Decision table of model values.

Universe of discourse U	Model attributes				Value v
	s_1	s_2	\dots	s_k	
1	Middle	Low	\dots	High	Low
2	Middle	High	\dots	Middle	Middle
\dots	\dots	\dots	\dots	\dots	\dots
l'	Low	High	\dots	High	High

The significance of model attribute s_g in Table 1 varies with decision attribute v . In order to determine the significance, we investigate how the decision table classification varies with the removal of the model attribute from the decision table. Generally, if model attribute s_g is deleted from condition attribute set S , then the impact of deleting s_g on the classification ability of S relative to decision attribute v increases with the value of $\gamma_{\text{IND}(S)}(v) - \gamma_{\text{IND}(S-s_g)}(v)$, namely, s_g becomes more significant for S relative to decision attribute v .

Definition 2. The significance of the model attribute s_g for the condition attribute set S relative to the decision attribute v is formulated as

$$\begin{aligned} \text{sig}(s_g, S; v) &= \gamma_{\text{IND}(S)}(v) - \gamma_{\text{IND}(S-s_g)}(v) \\ &= \frac{|\text{pos}_{(S)}(v)| - |\text{pos}_{(S-s_g)}(v)|}{|U|}, \quad g = 1, 2, \dots, k, \end{aligned} \quad (2)$$

where $\gamma_{\text{IND}(S)}(v)$ is the approximation quality of v by S , $\gamma_{\text{IND}(S-s_g)}(v)$ is the approximation quality of v by $S-s_g$, $\text{pos}_{(S)}(v)$ is the S -positive region of v , and $\text{pos}_{(S-s_g)}(v)$ is the $(S-s_g)$ -positive region of v .

Therefore, the weight of the model attribute s_g is obtained as follows:

$$w_g = \frac{\text{sig}(s_g, S; v)}{\sum_{g=1}^k \text{sig}(s_g, S; v)}. \quad (3)$$

3.2. Model Selection Based on Advantage Number Analysis. To improve the development efficiency, an optimal selection must be conducted before the model assignment to filter out parts of the preferred models for specific tasks, given that not all the models would be applied in all the tasks. In this case, the model selection uses an approach based on advantage number analysis.

According to Definition 1, the values of the l models M_1, M_2, \dots, M_l , for task T_j provide the value vector $\mathbf{V}_j = (v_{1j}, v_{2j}, \dots, v_{lj})^T$. Set the highest value component of \mathbf{V}_j as the ordinal number one, the second-highest value component as the ordinal number two, and so on. Further, \mathbf{V}_j can be converted in the ordinal number vector $\mathbf{R}_j = (r_{1j}, r_{2j}, \dots, r_{lj})^T$ of the values of l models for task T_j ,

whereby r_{ij} ($i = 1, 2, \dots, l$) denotes the order of model M_i in the model value list for task T_j .

Correspondingly, in the task set, n tasks produce n ordinal number vectors, and all the ordinal number vectors would constitute a model value ordinal number matrix with $l \times n$ dimensions.

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{l1} & r_{l2} & \dots & r_{ln} \end{bmatrix}. \quad (4)$$

Based on the advantage number analysis method, the respective advantage number of each element in matrix \mathbf{R} would be

$$a_{ij} = l + 1 - r_{ij}. \quad (5)$$

The ordinal number r_{ij} can be converted to an advantage number a_{ij} . Apparently, a lower ordinal number generates a higher model value on the specific task and a bigger advantage number. Therefore, the model value ordinal number matrix can be converted to a model value advantage number matrix.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{l1} & a_{l2} & \dots & a_{ln} \end{bmatrix}. \quad (6)$$

By summing up all the advantage numbers on each line in \mathbf{A} , the advantage number summation of the specific model corresponding to the line is obtained.

$$A_i = \sum_{j=1}^n a_{ij}, \quad i = 1, 2, \dots, l. \quad (7)$$

Upon selection of a proper threshold λ , all the $A_i \geq \lambda$ models are the assignable models selected from the model repository. Assuming that the number of the selected models is m , the value matrix of m selected models for n tasks of the task set can form an $m \times n$ dimensional matrix.

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \dots & \dots & \dots & \dots \\ v_{m1} & v_{m2} & \dots & v_{mn} \end{bmatrix}. \quad (8)$$

3.3. Establishment of a Multiobjective Model of Model Assignment. Without considering possible constraints on resources, the problem associated with the design-task-oriented model assignment in MBSE involves n tasks within a development project that applies an MBSE

mode. Precedence relations exist between some of the tasks that prohibit the onset of the task T_j ($j = 2, 3, \dots, n$) before all of its precedence tasks T_h ($h \in P_j$) are completed. Secondly, the model repository involves m selected models for all the specified tasks. Task T_j ($j = 1, 2, \dots, n$) has to choose one of the selected models to be performed, and the model cannot be ceased or changed to another model form during the task once the model is assigned selectively. The actual value of model M_i is different after it has been assigned to different tasks. Meanwhile, for task T_j , the cycle varies depending on the application of different models. Thus, the cycle matrix of

m selected models for n tasks in the task set can be expressed as

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix}. \quad (9)$$

The model assignment must meet the two objectives, namely, the minimization of the task set execution cycle and the maximization of the models' actual value summation.

A decision variable can be introduced as

$$x_{ijt} = \begin{cases} 1, & \text{model } M_i \text{ is assigned to task } T_j \text{ and task } T_j \text{ is accomplished at phase } t, \\ 0, & \text{others.} \end{cases} \quad (10)$$

Thus, the design-task-oriented model assignment in MBSE is presented as a multiobjective model according to

$$\min F_n = \sum_{i=1}^m \sum_{t=EF_n}^{LF_n} t \cdot x_{ijt}, \quad (11)$$

$$\max V = \sum_{i=1}^m \sum_{j=1}^n x_{ijt} v_{ij}, \quad (12)$$

such that

$$\sum_{i=1}^m \sum_{t=EF_j}^{LF_j} x_{ijt} = 1, \quad j = 1, 2, \dots, n, \quad (13)$$

$$\sum_{i=1}^m \sum_{t=EF_h}^{LF_h} t \cdot x_{iht} \leq \sum_{i=1}^m \sum_{t=EF_j}^{LF_j} (t - d_{ij}) x_{ijt}, \quad (14)$$

$$j = 2, 3, \dots, n; h \in \mathcal{P}_j,$$

$$x_{ijt} \in \{0, 1\}, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n; \\ t = EF_j, \dots, LF_j. \quad (15)$$

Equations (11) and (12) are objective functions that, respectively, denote the minimization of the task set execution cycle and the maximization of the models' actual value summation (after the models are assigned to the tasks). Equation (13) indicates that each task requires only a single model and can be executed only once. Equation (14) indicates that task T_j can only be started if all the precedence tasks T_h are accomplished. Equation (15) defines the range of values of the variable. Correspondingly, a solution of the multiobjective model is achieved after the determination of each model's assignment manner and each task's completion time F_j .

4. Algorithm for the Solution of the Multiobjective Model Assignment

The design-task-oriented model assignment problem is a multiobjective optimization problem. Many optimization algorithms have been developed to solve the optimization problem [29, 30], such as NSGA-II presented by Deb et al. [31, 32] and PSO developed by Kennedy and Eberhart [33]. The algorithm was proved to be superior to other evolutionary algorithms regarding the overall fitness [34]. In MBSE projects, the algorithm was applied to aviation [15], space [7, 35], software [36], and manufacturing [37]. In these applications, the researchers of NSGA-II achieved improvements.

This study introduces the DA-NSGA-II algorithm to solve the proposed design-task-oriented model assignment. The algorithm framework is shown in Figure 2 and the operating procedures are described below:

Procedure 1: establishment of the initial population. The codes of the chromosomes are generated based on the different task assignment manners of each selected model. The code length equals the task quantity n , and the code bits denote the corresponding tasks. The value range of each code bit ranges from 1 to m , while the actual value is decided by the model code that is being assigned to the specific task represented by code bits. According to the coding rule, the initial population is obtained to utilize the individuals that are randomly generated according to the population size to meet the constraints.

Procedure 2: the first fast nondominated sorting. Upon the calculation of each individual's execution cycle of the task set and the actual value summation of the models, the fitness values of the individuals are obtained. Each individual's ordinal number and crowding distance are then obtained by fast sorting the individuals in a non-domination manner based on the fitness value.

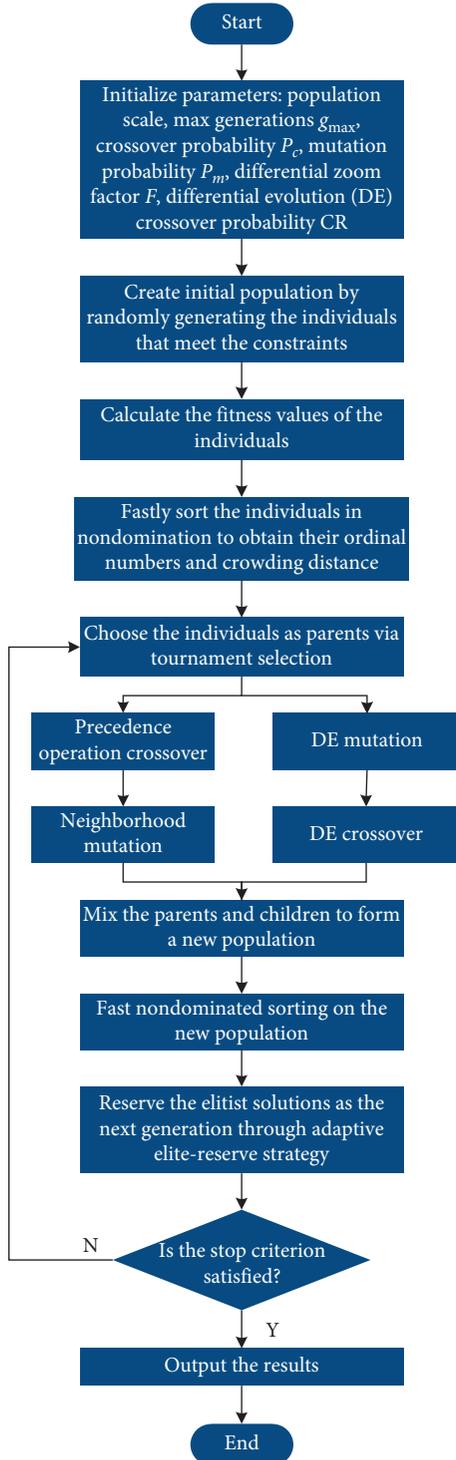


FIGURE 2: DA-NSGA-II framework.

Procedure 3: crossover and mutation. Consider some of the individuals selected via tournament selection as parents. First, the children with an equal number of the parents are achieved with precedence operation crossover (POX) and neighborhood mutation. To increase the diversity of the children, a differential evolution (DE) algorithm is used on the parents to generate

the second batch of children with an equal number of the parents.

Procedure 4: the second fast nondominated sorting. The two batches of children obtained based on crossover, mutation, and DE are introduced in the population, and each individual's ordinal number and crowding distance can then be obtained by executing the second fast nondominated sorting on the new population.

Procedure 5: we introduce an adaptive algorithm in the elite-reserve solution and reserve the elitist individuals from the lowest ordinal number to the highest one as the next generation until the number of individuals reaches the defined population size. In the cases of individuals with the same ordinal numbers, those with larger crowding distances are preferred to be reserved. The reservation rate is

$$\rho = \frac{1}{1 + e^{-t}}, \quad (16)$$

where t indicates the number of iterations.

Procedure 6: we output the results once the stop criterion is satisfied, which, in our case, is the maximum number of generations. Alternatively, the execution jumps to Procedure 3.

5. Case Study

5.1. *Assignment Solution of Models to the Design Task Set in a Phased-Array Radar Development Project.* The MBSE method was applied in a phased-array radar development project. The project's task set involved 14 tasks: T_1 for logical architecture decomposition, T_2 for the simulations of the main lobe and the side-lobe characteristics of the radar antenna, T_3 for interference suppression simulations, T_4 for clutter suppression simulations, T_5 for the assignment of component functions, T_6 for component interface definitions, T_7 for amplitude-phase consistency design, T_8 for radio frequency (RF) modeling, T_9 for RF simulations, T_{10} for scan matching simulations, T_{11} for small-scale modeling, T_{12} for small-scale simulations, T_{13} for radar cross section simulations, and T_{14} for virtual system integration analysis. The temporal relations among tasks are precedence restrictions (i.e., start-end relations) (see Figure 3 for the temporal relations).

There are 15 models whose disciplines match the task set. Each model involves five attributes, that is, model integrity (s_1), simulation operating efficiency (s_2), simulation confidence (s_3), model compatibility (s_4), and model interoperability (s_5). See Table 2 for the execution cycles of each task when separate models are applied.

The objective of the case is to find the optimal solution that assigns the models to the design tasks in the task set based on the proposed method. We first perform an optimal selection of the models. To calculate the model values, the weights of the model attributes have to be determined as a prerequisite. Selecting 12

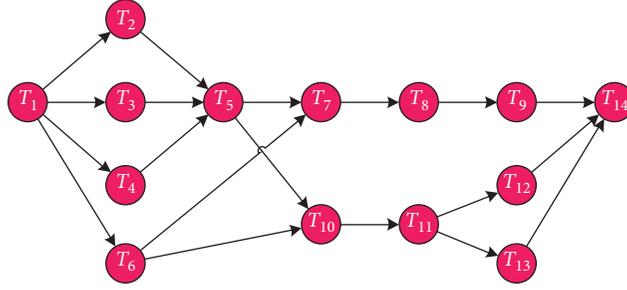


FIGURE 3: Temporal relations between model-based design tasks of a phased-array radar.

TABLE 2: Execution cycles of each task when separate models are applied (unit: day).

Model	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}
M_1	3	5	18	9	4	6	8	3	2	4	8	10	11	7
M_2	3	4	11	11	5	9	8	4	4	4	9	9	9	8
M_3	2	6	14	10	3	7	7	3	3	7	6	11	10	6
M_4	3	5	16	13	4	8	10	5	5	5	8	8	12	8
M_5	5	7	12	10	6	5	9	6	2	5	11	12	11	7
M_6	4	9	9	8	4	7	8	2	2	3	7	14	9	12
M_7	3	5	11	12	4	6	8	4	3	4	10	9	13	10
M_8	6	8	17	15	6	4	11	3	1	6	8	10	10	5
M_9	5	6	13	9	5	10	9	6	3	4	12	7	15	11
M_{10}	4	8	12	7	7	9	6	3	4	5	7	15	17	9
M_{11}	5	7	15	14	4	6	10	2	1	3	11	9	14	7
M_{12}	6	4	13	11	5	7	9	4	5	6	10	13	10	9
M_{13}	2	3	14	12	6	8	11	5	2	7	8	8	8	6
M_{14}	4	6	10	8	5	5	7	3	4	5	9	11	12	7
M_{15}	3	5	12	9	3	6	12	5	3	4	9	9	16	10

original data of model value assessment as the universe of discourse U , the conditional attribute set indicates the model attributes $\{s_1, s_2, s_3, s_4, s_5\}$, and the decision attribute set indicates the model value $\{v\}$, thus establishing the decision table with the model values, as shown in Table 3.

According to (2) and (3), the weights of the model attributes $s_1, s_2, s_3, s_4,$ and s_5 would be

$$\begin{aligned}
 w_1 &= \frac{\text{sig}(s_1, S; v)}{\sum_{g=1}^5 \text{sig}(s_g, S; v)} = \frac{0.5}{1.584} = 0.316, \\
 w_2 &= \frac{\text{sig}(s_2, S; v)}{\sum_{g=1}^5 \text{sig}(s_g, S; v)} = \frac{0.167}{1.584} = 0.105, \\
 w_3 &= \frac{\text{sig}(s_3, S; v)}{\sum_{g=1}^5 \text{sig}(s_g, S; v)} = \frac{0.167}{1.584} = 0.105, \\
 w_4 &= \frac{\text{sig}(s_4, S; v)}{\sum_{g=1}^5 \text{sig}(s_g, S; v)} = \frac{0.417}{1.584} = 0.263, \\
 w_5 &= \frac{\text{sig}(s_5, S; v)}{\sum_{g=1}^5 \text{sig}(s_g, S; v)} = \frac{0.333}{1.584} = 0.210.
 \end{aligned} \tag{17}$$

According to Definition 1, model values can be obtained for various tasks. For instance, the 15 model values for task T_1 are listed in Table 4.

Therefore, the value vector of the 15 models for task T_1 is $\mathbf{V}_1 = (5.099, 5.205, 4.889, 5.311, 3.835, 4.787, 5.731, 3.731, 4.679, 4.997, 3.945, 4.571, 5.627, 3.521, 4.995)^T$, and it would be transformed to the ordinal number vector $\mathbf{R}_1 = (5, 4, 8, 3, 13, 9, 1, 14, 10, 6, 12, 11, 2, 15, 7)^T$. According to (4), the ordinal number matrix \mathbf{R} composed of the ordinal number vectors of all the tasks from the task set can be formulated. According to (5), the advantage number matrix \mathbf{A} can be realized based on the conversion from the ordinal number matrix. Finally, based on (7), the advantage number summations of each model can be achieved, as listed in Table 5.

According to the experience, choosing the threshold $\lambda = 106$, the assignable selected model set in the model repository shall be $\{M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}\}$. The value matrix of the 10 models in the selected model set for the 14 tasks in the task set can be expressed as Table 6.

We attempted to solve the assignment of each selected model to the task set by utilizing DA-NSGA-II according to the objective functions of (11) and (12), based on the assumption of the following parameters: 200 for the initial

TABLE 3: Decision table with a phased-array radar’s model values.

Universe of discourse U	Conditional attributes S					Decision attribute D
	s_1	s_2	s_3	s_4	s_5	v
1	Middle	Low	Middle	Middle	Middle	Low
2	Middle	High	Middle	Middle	Middle	Middle
3	Middle	High	Middle	Middle	High	High
4	Middle	Low	Low	Middle	Low	Low
5	Middle	Low	Middle	Middle	Low	Middle
6	Middle	High	Middle	High	Middle	High
7	Low	High	High	Middle	Low	Middle
8	Low	Low	Middle	Middle	Low	Low
9	High	High	Middle	Middle	Middle	High
10	Middle	Low	Middle	High	Low	Middle
11	Middle	Low	Middle	Low	Low	Low
12	Middle	High	High	Middle	Low	High

TABLE 4: Model values for task T_1 .

Model	s_1 0.316	s_2 0.105	s_3 0.105	s_4 0.263	s_5 0.210	v_{i1}
M_1	3	7	5	7	5	5.099
M_2	7	5	7	1	7	5.205
M_3	3	3	7	7	5	4.889
M_4	5	5	3	7	5	5.311
M_5	1	9	7	3	5	3.835
M_6	7	1	5	5	3	4.787
M_7	5	3	1	7	9	5.731
M_8	3	1	9	1	7	3.731
M_9	5	7	9	3	3	4.679
M_{10}	9	5	3	1	5	4.997
M_{11}	7	3	1	1	5	3.945
M_{12}	1	3	7	5	9	4.571
M_{13}	5	1	1	9	7	5.627
M_{14}	3	3	9	1	5	3.521
M_{15}	5	9	1	5	5	4.995

TABLE 5: Advantage number summations of each studied model.

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
108	140	99	129	110	131	98	109	119	101	107	132	118	85	105

TABLE 6: Value matrix of the selected models for all tasks.

Model	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}
M_1	5.099	3.309	5.209	3.523	6.155	4.573	4.047	4.575	5.523	2.155	5.099	6.153	1.839	6.469
M_2	5.205	5.311	5.417	5.835	4.785	6.151	4.467	4.891	4.681	4.787	4.781	5.835	7.203	6.151
M_4	5.311	5.941	8.151	4.993	3.733	3.103	5.839	8.151	6.361	6.995	3.311	4.047	3.521	3.101
M_5	3.835	4.891	3.627	7.205	6.575	4.259	2.575	3.733	5.627	5.413	4.259	3.627	4.679	5.519
M_6	4.787	4.997	4.891	3.521	4.785	7.729	1.841	6.783	4.997	2.259	7.835	6.993	5.523	6.571
M_8	3.731	2.889	5.627	3.733	4.467	5.419	3.311	1.209	5.519	7.731	6.151	6.677	5.627	2.679
M_9	4.679	5.519	4.783	4.465	6.363	4.783	5.731	4.891	5.207	5.417	4.471	6.257	1.419	4.261
M_{11}	3.945	5.205	6.365	6.255	3.735	6.361	3.101	6.889	4.785	1.735	2.577	2.995	5.625	4.365
M_{12}	4.571	3.521	6.153	5.419	5.101	4.045	6.571	3.209	4.049	4.573	7.309	7.099	5.099	7.415
M_{13}	5.627	5.523	3.101	4.573	5.521	6.889	3.313	4.155	4.993	4.787	4.365	6.677	4.363	2.261

TABLE 7: Objective values corresponding to the Pareto optimal solution set with DA-NSGA-II.

Object	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7	Solution 8	Solution 9	Solution 10	Solution 11	Solution 12	Solution 13	Solution 14	Solution 15	Solution 16	Solution 17	Solution 18
Execution cycle	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
Value summation	77.6	81.7	85.4	88.6	90.1	92.3	93.9	94.7	95.2	95.6	96.1	96.3	97.2	97.9	98.1	98.9	99.1	99.3

TABLE 8: Models for various tasks demonstrated by solution 10 in Table 7.

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}
M_{13}	M_{13}	M_6	M_5	M_1	M_6	M_{12}	M_4	M_4	M_8	M_6	M_2	M_2	M_{12}

population size, 200 for maximum generations, probability of crossover $P_c=0.9$, probability of mutation $P_m=0.1$, differential zoom factor $F=0.5$, and probability of DE crossover $CR=0.7$. A set of Pareto optimal solutions were obtained, and the objective values are presented in Table 7.

In the phased-array radar development project, we can select one of the solutions from the Pareto optimal solution set by considering the practical constraint of the execution cycle and the expectation of the of the model value summation to determine the eventual model assignment scheme. For example, the assignment scheme demonstrated by solution 10 is listed in Table 8.

5.2. Algorithms Comparison

5.2.1. Performances Comparison. The traditional NSGA-II and PSO have also been used to solve the model assignment model to verify the performance of the proposed algorithm. The same parameters are preset in both NSGA-II and DA-NSGA-II. The initial population size and maximum generations of PSO are equal to those of DA-NSGA-II, with the remaining PSO parameters set as follows: acceleration constants $c_1=0.8$ and $c_2=0.8$; inertia weight $\omega_{\max}=1.2$ and $\omega_{\min}=0.1$. The comparison diagram of the Pareto fronts generated by the three algorithms is shown in Figure 4.

From Figure 4, we can always find superior solutions in the Pareto front of DA-NSGA-II by comparing to the Pareto front of the traditional NSGA-II and PSO. This proves that the convergence of DA-NSGA-II is superior to that of NSGA-II and PSO.

To conduct a quantitative evaluation on the performance of DA-NSGA-II, the study applied the S- and M-measures as the evaluating indicators. Based on the solutions of the model assignment with DA-NSGA-II, NSGA-II, and PSO, Table 9 presents a comparison of S- and M-measures of the three Pareto optimal solutions achieved by executing separately the three algorithms 10 different times.

Table 9 demonstrates that the mean value of the S-measure in the DA-NSGA-II case is lower than those associated with NSGA-II and PSO, and the mean value of the M-measure in the DA-NSGA-II case is higher than those obtained from the execution of NSGA-II and PSO. It also shows that the standard deviations of the S- and M-measures in the DA-NSGA-II case are lower than those in the NSGA-II and PSO cases. These results indicate that the distribution uniformity and the range of the solution outcomes generated by the DA-NSGA-II algorithm are superior to those of the NSGA-II and PSO algorithms-based solutions, and the outputs of DA-NSGA-II are more stable than those of NSGA-II and PSO.

Figure 5 compares the execution cycles and model value summations outputted by DA-NSGA-II, NSGA-II, and PSO

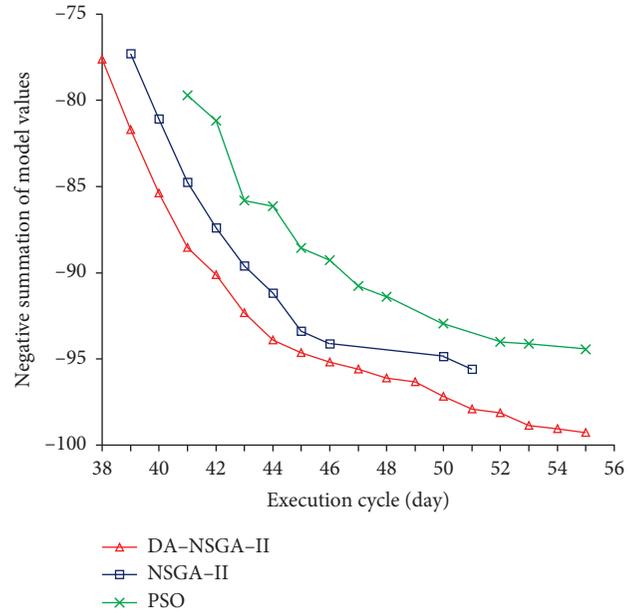


FIGURE 4: Comparison of the Pareto fronts generated by DA-NSGA-II, NSGA-II, and PSO.

from generations 1 to 200. The value of each execution cycle is the minimum execution cycle in the Pareto solution set outputted by the algorithms at each generation (Figure 5(a)). Each model value summation is the maximum summation of the model values in the Pareto solution set outputted by the algorithms at each generation (Figure 5(b)). The execution cycle and model value summations achieved by DA-NSGA-II begin to converge before generation 20, while those of DA-NSGA-II begin to converge at generations 21 and 37, respectively. PSO is not able to attain the optimal values until generations 101 and 135, respectively. These results indicate the faster convergence rate of DA-NSGA-II compared to those of NSGA-II and PSO.

5.2.2. Optimizing Effects Comparison. In this section, we compare the optimizing effects via DA-NSGA-II, NSGA-II, and PSO. First, the execution cycles of the task set corresponding to the model assignment schemes optimized by the three algorithms are compared. The comparison is performed between solutions with equal model value summations. However, from DA-NSGA-II Pareto optimal solution set, we cannot determine a solution with model value summation exactly equal to that of the Pareto optimal solution from NSGA-II or PSO. Therefore, solution sets A and B in the Pareto front generated by DA-NSGA-II are calculated via interpolation. The model value summations corresponding to the solutions in solution set A (B) are equal

TABLE 9: Comparison of the performances of the three algorithms.

Evaluation index	DA-NSGA-II		NSGA-II		PSO	
	Mean value	Standard deviation	Mean value	Standard deviation	Mean value	Standard deviation
S-measure	0.8201	0.1602	0.8784	0.4468	1.2693	0.8716
M-measure	23.7390	2.5971	21.0179	3.6404	23.2693	3.3199

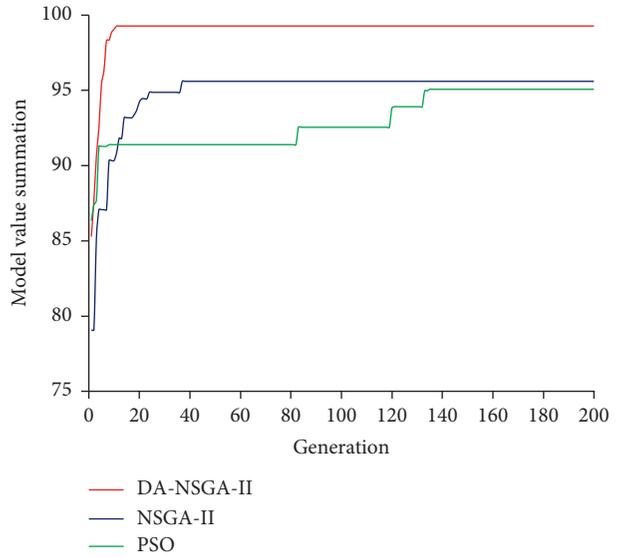
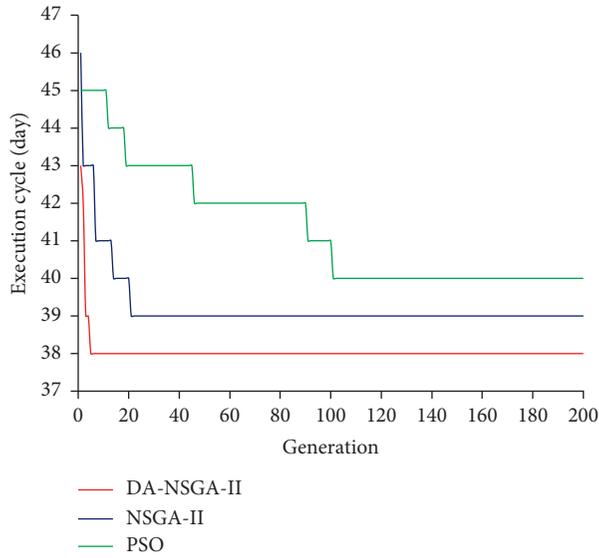


FIGURE 5: Comparison of the objective values outputted by DA-NSGA-II, NSGA-II, and PSO across multiple generations. (a) Execution cycles of the three algorithms from generations 1 to 200. (b) Model value summations of the three algorithms from generations 1 to 200.

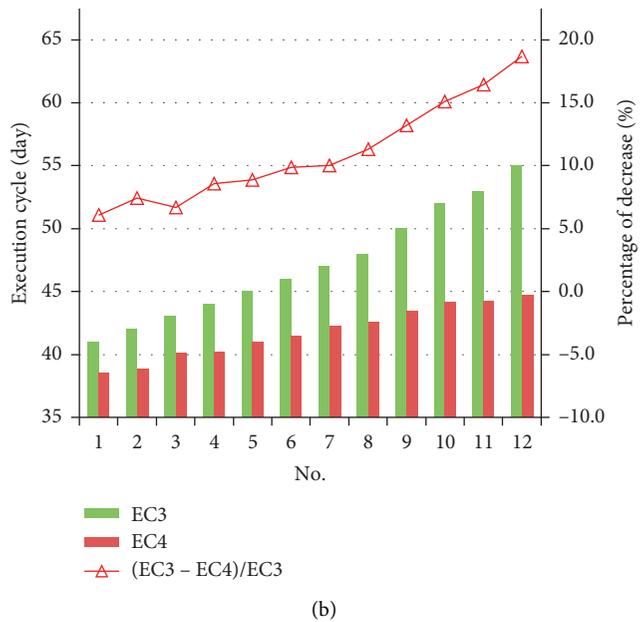
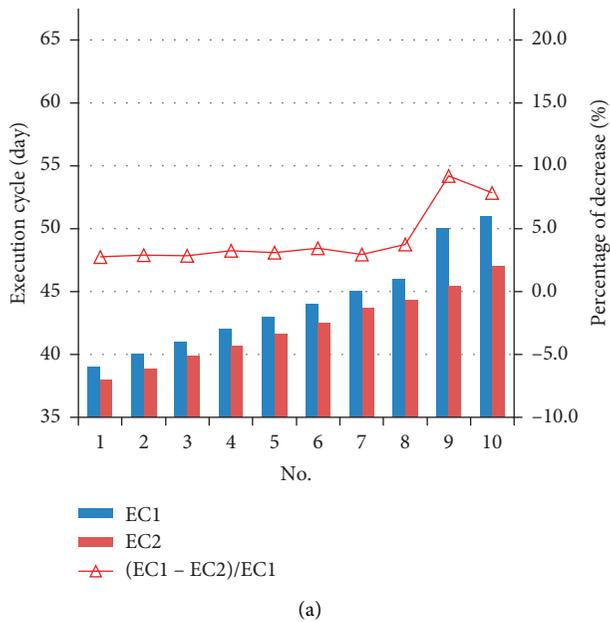


FIGURE 6: Comparison of the task set execution cycles optimized by the three algorithms, assuming equal model value summations. (a) Comparison of EC2 obtained by DA-NSGA-II and EC1 obtained by NSGA-II. Numbers 1–10 indicate the model value summations in the Pareto optimal solution set from NSGA-II of 77.3, 81.1, 84.8, 87.4, 89.6, 91.2, 93.4, 94.1, 94.9, and 95.6, respectively. (b) Comparison of EC4 determined by DA-NSGA-II and EC3 determined by PSO. Numbers 1–12 indicate the model value summations in the Pareto optimal solution set from PSO of 79.7, 81.2, 85.8, 86.1, 88.6, 89.3, 90.8, 91.4, 93.0, 94.0, 94.1, and 94.4, respectively.

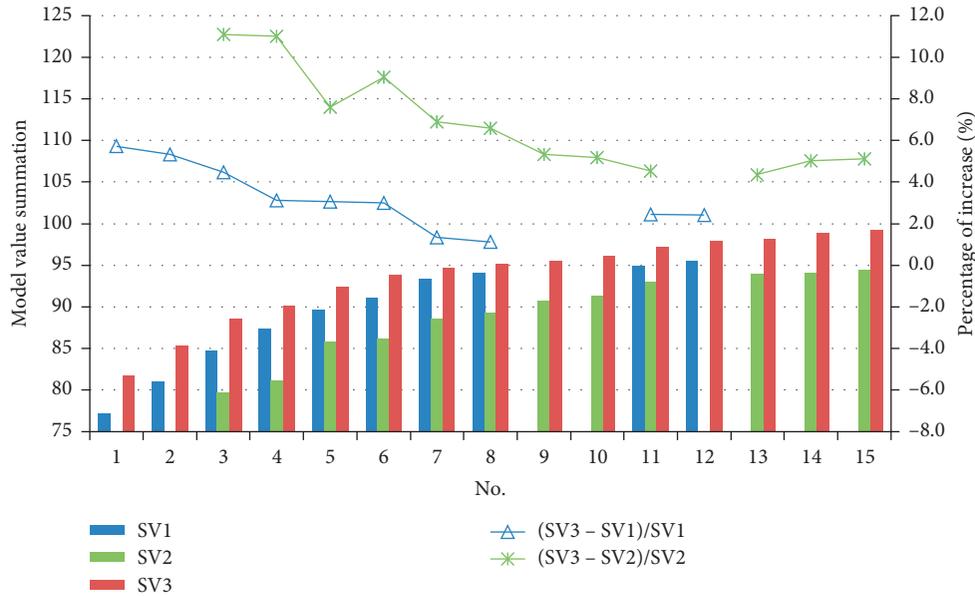


FIGURE 7: Comparison of the model value summations optimized by the three algorithms under equal task set execution cycles. Numbers 1–15 indicate the task set execution cycles of 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 50, 51, 52, 53, and 55 (unit: day), respectively.

to those of the Pareto optimal solutions based on NSGA-II (PSO). We denote the execution cycle of the task set related to the Pareto optimal solution from NSGA-II, set A, PSO, and set B as EC_1 , EC_2 , EC_3 , and EC_4 , respectively. Figure 6 presents the comparison between EC_1 and EC_2 and between EC_3 and EC_4 , under the condition that the corresponding summations of the model values are equal.

The bar chart in Figure 6(a) compares EC_1 and EC_2 with equal model value summations from the Pareto optimal solution sets generated by DA-NSGA-II and NSGA-II, respectively, while the line graph presents the percentage decrease of EC_2 compared to EC_1 . EC_2 is 2.8–9.2% lower than EC_1 . Similarly, the bar chart in Figure 6(b) compares EC_3 and EC_4 with equal model value summations from the Pareto optimal solution sets generated by DA-NSGA-II and PSO, respectively, while the line graph presents the percentage decrease of EC_4 compared to EC_3 . EC_4 is 6.1–18.7% lower than EC_3 . Thus, the design efficiency of the model assignment solution optimized by DA-NSGA-II is higher than those optimized by NSGA-II and by PSO. The efficiency improved on average by 4.2% and 11.0%, respectively.

Second, the model value summations corresponding to the model assignment solutions optimized by the three algorithms are compared. We define the model value summation of the Pareto optimal solution generated by NSGA-II, PSO, and DA-NSGA-II as SV_1 , SV_2 and SV_3 , respectively. Figure 7 compares SV_1 , SV_2 , and SV_3 under the condition of equal task set execution cycles.

The bar chart in Figure 7 compares SV_1 , SV_2 , and SV_3 for equal task set execution cycles from the Pareto optimal solution sets obtained by the three algorithms, while the line graph presents the percentage increase of SV_3 compared to SV_1 and that of SV_3 compared to SV_2 . SV_3 is 1.1–5.7% greater than SV_1 , with an average increase of 3.2%, while SV_3 is 4.4–11.1% greater than SV_2 , with an average increase of 6.8%. Therefore, the model assignment solution optimized by DA-

NSGA-II promotes the design task set to a greater extent compared with those optimized by NSGA-II and PSO.

6. Conclusions

The proposed design-task-oriented model assignment framework that involves model value selection, multi-objective model establishment, and the multiobjective optimization algorithm provides a solution for the problem of model assignment in the model repository to the design tasks in MBSE.

In MBSE, design-task-oriented model assignment based on model value quantification involves the use of multi-objective optimization. This study applies an advantage-number-based analytical technique to quantify the models' values and consequently obtains the value matrix of the models. The goals of the multiobjective model of model assignment, which is established based on the models' cycle and value matrices on the specified tasks, are the minimization of the task set execution cycle and the maximization of the models' actual value summation.

The proposed DA-NSGA-II algorithm increases the diversity of the population and reserves more elite solutions based on the introduction of the DE algorithm and the adaptive elite-reserve solution. The convergence, distribution uniformity, and ranges of the solution outcomes of the algorithm are superior to those of the traditional NSGA-II and PSO. Moreover, the algorithm is more conducive to improving the design efficiency and effect than NSGA-II and PSO. The algorithm is therefore favorable for the solution of problems that involve design-task-oriented model assignments.

Further research should focus on the following two additional perspectives. (1) In the current study, the establishment of a multiobjective model assumed a precondition of unlimited resources. However, resources that

include human resources, cash, cost, and other factors are constrained in a realistic product development project. Future research should execute the model establishment over resource-constrained conditions to further enhance and broaden the applicability of the algorithm. (2) For an MBSE-compliant product development project, model scheduling software is capable of improving the usage effects of the models and the development cycles. According to the findings of this study, one can develop a model scheduling software based on the proposed multiobjective model and DA-NSGA-II algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (grant no. 51575274).

References

- [1] K. Bhasin, P. Barnes, J. Reinert, and B. Golden, "Applying model based systems engineering to NASA's space communications networks," in *Proceedings of the 7th Annual IEEE International Systems Conference*, pp. 325–330, Orlando, FL, USA, April 2013.
- [2] S. M. Simi, S. P. Mulholland, and W. G. Tanner, "TES-SAVi AWESUM model-based systems engineering (MBSE) for FACE applications," in *Proceedings of the 2014 IEEE Aerospace Conference*, p. 17, Big Sky, MT, USA, March 2014.
- [3] E. M. Kraft, "HPCMP CREATE™-AV and the air force digital thread," in *Proceedings of the 53rd AIAA Aerospace Sciences Meeting*, p. 13, Kissimmee, FL, USA, January 2015.
- [4] J. J. Zhang, Z. Liu, F. Li et al., "Employing model-based systems engineering (MBSE) on a civil aircraft research project: a case study," in *Proceedings of the 2018 Asia-Pacific International Symposium on Aerospace Technology*, pp. 2178–2186, Chengdu, China, October 2018.
- [5] P. M. Fischer, D. Lüdtkke, C. Lange, F.-C. Roshani, F. Dannemann, and A. Gerndt, "Implementing model-based system engineering for the whole lifecycle of a spacecraft," *CEAS Space Journal*, vol. 9, no. 3, pp. 351–365, 2017.
- [6] D. Kaslow and A. M. Madni, "Validation and verification of MBSE-compliant CubeSat reference model," *Disciplinary Convergence in Systems Engineering Research*, A. M. Madni, B. Boehm, R. G. Ghanem et al., Eds., pp. 381–393, Springer, Cham, Switzerland, 2018.
- [7] J. Stern, S. Wachtel, J. Colombi, D. Meyer, and R. Cobb, "Multiobjective optimization of geosynchronous earth orbit space situational awareness systems via parallel executable architectures," in *Proceedings of the Conference on Systems Engineering Research (CSER 2017)*, pp. 599–615, Redondo Beach, CA, USA, March 2017.
- [8] Q. Do, S. Cook, and M. Lay, "An investigation of MBSE practices across the contractual boundary," *Procedia Computer Science*, vol. 28, pp. 692–701, 2014.
- [9] C. Wang, "MBSE-compliant product lifecycle model management," in *Proceedings of the 14th Annual Conference System of Systems Engineering*, pp. 248–253, Anchorage, AK, USA, 2019.
- [10] International Council on Systems Engineering (INCOSE), *Systems Engineering Vision 2020, Version 2.03*, INCOSE, San Diego, CA, USA, 2007.
- [11] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*, Morgan Kaufmann Publishers, Waltham, MA, USA, 2012.
- [12] G. Barbieri, C. Fantuzzi, and R. Borsari, "A model-based design methodology for the development of mechatronic systems," *Mechatronics*, vol. 24, no. 7, pp. 833–843, 2014.
- [13] C. Steimera, J. Fischerb, and J. C. Aurich, "Model-based design process for the early phases of manufacturing system planning using SysML," in *Proceedings of the 27th CIRP Design Conference*, pp. 163–168, Cranfield, UK, May 2017.
- [14] K. Kübler, S. Scheifele, C. Scheifele, and O. Riedel, "Model-based systems engineering for machine tools and production systems (model-based production engineering)," *Procedia Manufacturing*, vol. 24, pp. 216–221, 2018.
- [15] D. J. L. Siedlak, O. J. Pinon, P. R. Schlais, T. M. Schmidt, and D. N. Mavris, "A digital thread approach to support manufacturing-influenced conceptual aircraft design," *Research in Engineering Design*, vol. 29, no. 2, pp. 285–308, 2018.
- [16] K. Hampson, "Technical evaluation of the systems modeling language (SysML)," *Procedia Computer Science*, vol. 44, pp. 403–412, 2015.
- [17] S. Gao, W. Cao, L. Fan, and J. Liu, "MBSE for satellite communication system Architecting," *IEEE Access*, vol. 7, pp. 164051–164067, 2019.
- [18] H. Xia, J. Jiao, and J. Dong, "Extend UML based timeliness modeling approach for complex system," in *Proceedings of the 2018 International Conference on Mathematics, Modeling, Simulation and Statistics Application*, p. 6, Shanghai, China, December 2018.
- [19] M. Challenger, G. Kardas, and B. Tekinerdogan, "A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems," *Software Quality Journal*, vol. 24, no. 3, pp. 775–795, 2016.
- [20] P. Roques, "MBSE with the ARCADIA method and the capella tool," in *Proceedings of the 8th European Congress on Embedded Real Time Software and Systems*, p. 10, Toulouse, France, January 2016.
- [21] A. Soltan, S. Addouche, M. Zolghadri, M. Barkallah, and M. Haddar, "System Engineering for dependency analysis - a Bayesian approach: application to obsolescence study," *Procedia CIRP*, vol. 84, pp. 774–782, 2018.
- [22] Y. Ye and G. Fischer, "Reuse-conducive development environments," *Automated Software Engineering*, vol. 12, no. 2, pp. 199–235, 2005.
- [23] U. Shani, "Can ontologies prevent MBSE models from becoming obsolete?" in *Proceedings of the 2017 Annual IEEE International Systems Conference*, p. 8, Montreal, Canada, April 2017.
- [24] H. Zhao, J. Liang, X. Yin et al., "Domain-specific ModelWare: to make the machine learning model reusable and reproducible," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 2, Oulu, Finland, October 2018.

- [25] B. Hamid, "Assessment of the SEMCO model-based repository approach for software system engineering," in *Proceedings of the 7th International Conference on Model and Data Engineering*, pp. 111–125, Barcelona, Spain, October 2017.
- [26] C. Ponsard, R. Darimont, and M. Touzani, "Robust design of a collaborative platform for model-based system engineering: experience from an industrial deployment," in *Proceedings of the 9th International Conference on Model and Data Engineering*, pp. 333–347, Toulouse, France, October 2019.
- [27] B. Hamid, "A model repository description language-MRDL," in *Proceedings of the 15th International Conference on Software Reuse*, pp. 350–367, Limassol, Cyprus, June 2016.
- [28] Z. Pawlak, "Rough sets and decision tables," in *Proceedings of the 5th Symposium on Computation Theory*, pp. 187–196, Zaborów, Poland, December 1984.
- [29] F. P. Mahdi, P. Vasant, V. Kallimani, J. Watada, P. Y. S. Fai, and M. Abdullah-Al-Wadud, "A holistic review on optimization strategies for combined economic emission dispatch problem," *Renewable and Sustainable Energy Reviews*, vol. 81, no. 2, pp. 3006–3020, 2018.
- [30] J. M. Granado-Criado, S. Santander-Jiménez, M. A. Vega-Rodríguez, and Á. Rubio-Largo, "A multi-objective optimization procedure for solving the high-order epistasis detection problem," *Expert Systems with Applications*, vol. 142, Article ID 113000, 2020.
- [31] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pp. 849–858, Paris, France, September 2000.
- [32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [33] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Network*, pp. 1942–1948, Perth, Australia, November 1995.
- [34] Y. Li, T. Yue, S. Ali, and L. Zhang, "Zen-ReqOptimizer: a search-based approach for requirements assignment optimization," *Empirical Software Engineering*, vol. 22, no. 1, pp. 175–234, 2017.
- [35] C. J. Lowe and M. Macdonald, "Rapid model-based interdisciplinary design of a CubeSat mission," *Acta Astronautica*, vol. 105, no. 1, pp. 321–332, 2014.
- [36] S. Rahmoun, A. Mehiaoui-Hamitou, E. Borde, L. Pautet, and E. Soubiran, "Multi-objective exploration of architectural designs by composition of model transformations," *Software & Systems Modeling*, vol. 18, no. 1, pp. 107–127, 2019.
- [37] R. Wang and C. H. Dagli, "Computational system architecture development using a holistic modeling approach," *Procedia Computer Science*, vol. 12, pp. 13–20, 2012.