Accounting for “Mission” During Co-optimization of System Designs

Tianlei Zhang, Grant Conklin, Yucheng Zhang, and Roger A. Dougal
Dept. of Electrical Engineering
University of South Carolina
Columbia, SC, USA
zhangt@email.sc.edu

Abstract*— A simulation-based approach to optimal design of systems requires collective solution to three related problems: selection of the best equipment, selection of the best system configuration, and optimal operation of the system in a mission-oriented sense. We solve these three problems by applying an improved Particle Swarm Optimization (PSO) algorithm on top of a toolbox that rapidly instantiates system models from high level specifications and system templates. The improved PSO more efficiently handles the involvement of discrete binary variables in the objective functions with universal constraints, more effectively avoids premature convergence, and yields a more accurate search for the global optimum solution. Our simulation-component-based co-optimization approach to system designs is illustrated by applying it to the design of an electric ship power system while accounting for the unit commitment problem during a set of missions. First, we prove the efficacy of the new PSO algorithm on one candidate system, and then we apply the new PSO method to evaluate and compare three candidate designs. The improved PSO shows that the system should actually consume up to 52% less fuel than is predicted by a simpler evaluation that invokes a proportionally-distributed power alignment; this is a much better measure of the true system performance. As compared with another validated hybrid PSO-GA algorithm, the improved PSO consistently finds an optimum operating point that yields up to 2.4% better fuel efficiency, and it improves the reliability of solutions by up to 32%. Next, three candidate designs are evaluated under optimal operating conditions, for a given mission profile, and compared in terms of their best performance. The simulation results successfully determine the best genset combination from the competing designs.

Keywords- system co-optimization; mission-oriented; Particle Swarm Optimization; unit commitment; electric ship

I. Introduction

Optimal design of systems poses three related problems: Which equipment to select? How to connect that equipment into a best system configuration? How to best control the equipment based on given mission? To achieve an optimal solution, these questions cannot be answered separately; nor even sequentially, but instead they must be answered collectively [1]. Consider the case of a stand-alone power system which, in various forms, could serve a remote community, a ship, or even a railroad engine. A simulation-based design tool for this problem should allow a designer to select from a range of equipment (e.g. generators of various power ratings, with various prime movers), to configure that equipment into a variety of systems (e.g. radial- or ring-bus networks), and to compute the performance of the equipment over a variety of missions (load power demand profiles), from which comparisons of performance (e.g. fuel consumption) could be made. The many permutations of equipment, configuration, and control strategies during missions make this a large co-optimization problem. In the optimization literature, two different system design approaches for this co-optimization problem have been described: one is pure-equation-based and the other is simulation-component-based. The former concurrently solves the three questions by a group of equations, which consist of the descriptions of all components in a system configuration (e.g. in terms of dynamics, dimensions or economic cost etc.) and the cost functions that define the design goals [2][3][4]. The latter constructs a complete system from ready-made components and configuration types in a simulation software tool and then tries to optimize the design variables [6][7]. The advantages and disadvantages of these two approaches are compared in TABLE I.

We employ the second approach – simulation-component-based – because, as indicated in the table, it results in more rapid, accurate and feasible solutions than the pure-equation-based approach. In order to address the 4th concern in the table, our optimization routine makes use of a recently developed simulation toolbox [15] that automatically instantiates a fully-configured system model based on high level descriptors of the system (such as number and ratings of generators number and ratings of load zones, power bus classification, etc). This greatly simplifies and speeds the task of assembling system models of candidate designs that will be evaluated after optimizing the high-level system controls. However, there are still two challenges in terms of system optimization strategy. One is that we lack of an efficient method to integrate certain optimizer with the simulation tool for the co-optimization process. The other is that existing optimization algorithms have difficulty tackling the characteristics of the cost function for system designs which contain discrete variables representing, for example, “ON” or “OFF” modes of operation, or indicating their discontinuous effective working intervals. Therefore, the main contribution in this paper is the development of an improved universal heuristic optimization algorithm which successfully integrated with the system definition toolbox to address these two challenges.

Typically heuristic techniques are selected for real system design problems over conventional gradient-based methods because heuristic techniques do not impose requirements that the systems be differentiable or continuous; there are no limits on assumptions regarding the objective function and

The authors acknowledge support from the US Office of Naval research, under grants N00014-08-1-0080 and N00014-07-1-0686.

978-1-4673-0750-5/12/$31.00 ©2012 IEEE

TABLE I. Comparison between Pure-Equation-Based and Simulation-Component-Based co-optimization approach for system designs

	Concern	Pure-Equation-Based		Simulation-Component-Based	
1	Sufficient access to the knowledge defining the system design problem for system engineers	✘	Requires expertise in multidisciplinary research domains; Limited by proprietary functions protected in industry.	✓	Provides ready-made modules maturely developed for general application purposes.
2	Error-proof process of defining the systems	✘	Uses different level-of-detailed equations for the same component as requested in various systems; Hard to validate the accuracy, compatibility and convergence when a group of equations combined.	✓	Guarantees the correctness of the pre-modeled components and configuration for universal purposes from various tests and validations.
3	Feasibility of Solutions	✘	Only valid for deciding continuous variables of systems, otherwise the products with the optimal design variables may not exist in the market	✓	Fits the real world cases because it guarantees the discrete selection of real existing components and configurations.
4	Flexibility in defining the design problems in various analyses	✓	Easy to update equations and constraints for describing the corresponding analyses (e.g. time-domain dynamic analyses, economic-domain analyses, or frequency-domain analyses etc.)	✘	Lacks flexibility: may require programming skills to recode the model, the uses of another software tool, or translation of simulation data from one tool to another
5	Optimization of control strategies	✓	Easy to apply certain optimization algorithm directly to the equations	✘	Seldom contains embedded universal optimization methodology for general purposes; Lacks efficient method to integrate external optimizer to the simulation software tool.

constraints; and the population-based search algorithms are less likely to become trapped on local minima [8].

Many heuristic methods are efficient in solving nonlinear, non-differentiable optimization problems. Some such methods considered are genetic algorithm (GA) [10], evolutionary programming (EP) [9], simulated annealing (SA) [10], bio-geography based optimization (BBO), gravitational search intelligence (GSI) [11] and Particle Swarm Optimization (PSO) [12]. However, most of these methods have drawbacks. GA, EP and SA do not always guarantee globally optimal solutions: GA suffers from complicated encoding and decoding schemes and degraded efficiency in applications where the design parameters are highly correlated; EP converges to near optimum rather slow due to its mutation and selection schemes and can get trapped in sub-optimal states for large scale complex problems; and SA is very time-consuming and difficult to tune the proper annealing schedule. BBO is time consuming in tuning parameters for complex systems. GSI's performance starts degrading largely when iteration numbers increase.

In contrast, PSO outperforms the aforementioned algorithms in several aspects [13]. PSO retains the advantages of population-based algorithms, being less susceptible to getting trapped in local minima. It balances the global and local exploration such that it converges to the global optimum in shorter time. It is easy to implement with basic mathematics and a limited number of parameters. Its performance does not depend on a user defined initial point for iteration. Additionally, it can be improved through integrating with other optimization techniques to solve a wide range of problems, e.g. mixed integer problems, multi-objective problems, objective function with stochastic nature, and problems with time-changing global optima. However, existing PSOs suffer from problem-dependent constraint handling method and premature convergence when dealing with discrete variables. Therefore, we propose an improved PSO that improves performance in avoiding premature convergence by adding a "mutation" operator and an "archive" vector for efficiently handling universal constrained cost functions.

In the next section, we describe how we integrated the system modeling toolbox with the PSO-based optimizer. Following that, in Section III we elaborate our improved PSO and methodology, then in Section IV we describe the mission-dependent unit commitment problem as related to operation of generating plants in an electric ship. Finally, in Section V we show and analyze the results of our optimization.

II. Integration of the Toolbox and Optimizer

We choose to use Virtual Test Bed (VTB) to demonstrate our optimization method because of our familiarity with its application programming interface (API). VTB comprises a suite of software tools for virtual prototyping of multi-disciplinary dynamic systems and it contains a suitable model library [14]. However, this work can be implemented in other software tools having an appropriate API.

We developed a user interface to parameterize and control the system assembly toolbox to realize the rapid system construction as the first step of co-optimization in VTB, as shown in Fig. 1. The interface consists of the schematic window where a user draws out and parameterizes the system blocks, the mission profile editor where discrete mission segments are defined, and the analyses selection window where the user can choose any number of preloaded analyses types and add them to the system model.

The procedures to realize the co-optimization process using the integrated system definition toolbox and optimizer algorithms are indicated by the data flow arrows in Fig. 1. The steps are elaborated as follows:

Step 1: The user determines the feasible system components and topologies (e.g. ring-, star-, and split-bus network etc., saved as individual files for future reuse) from available resources in software component library, loads

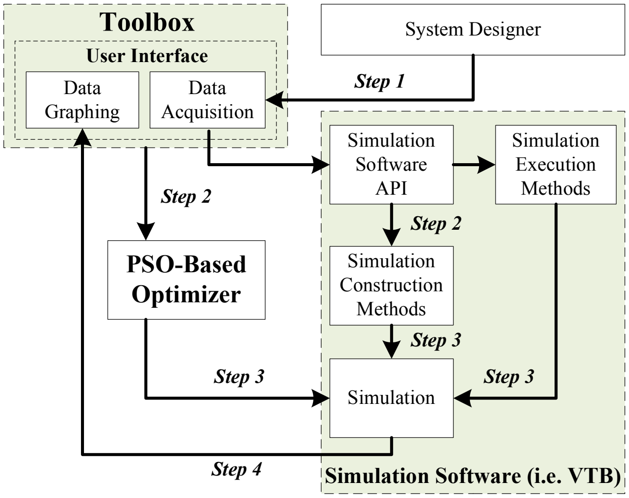

Figure 1. The data flow diagram of the simulation-component-based co-optimization system design approach

defined mission profile, and selects the types of analyses for particular design goals. The user also has to parameterize the components and create new topologies in this step, if necessary.

Step 2: Based upon the information from Step 1, the toolbox generates all feasible system designs, which are the candidates of the optimal solution to the final system design, into simulations whose results satisfy the design analyses requested by the user, and executes the optimizer to calculate the optimal control variables (e.g. how many generators are in service, and how much power should be committed from each generator to satisfy the load demands) for each candidate in terms of the same user defined control goals (the expression of cost function may vary for individual candidates due to their different components and configurations) based upon each mission segment.

Step 3: The toolbox invokes the simulation solver through software API to execute simulations of all the candidates, with their corresponding optimal control variables, to sequentially get the individual dynamic system performance for each candidate system. (This step could be parallelized in a many-processor environment).

Step 4: The user interface collects and displays the simulation results of all the candidates, for each of the requested analyses, for comparison and analysis.

III. The Improved Optimization Algorithm

Original PSO [12] was developed to perform as a flexible population-based stochastic search mechanism in order to combine and balance the global and local exploration capacities. The particle populations in PSO heuristically converge to the global optimum by learning from their own best previous experience and by communicating with each other to learn the best experience so far of the overall population. The relationship between global exploration and local exploitation in PSO is realized by the current position of each particle, x_d, the current velocity of each particle, v_d, the best position found so far by all particles, known as the *global best*, x_{gbest}, and the best history position of the individual particle, known as the *personal best*, x_{pbest}:

$$\begin{aligned} v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot U(0,1) \cdot \left[x_{gbest} - x_{id}(t) \right] \\ + c_2 \cdot U(0,1) \cdot \left[x_{pbest} - x_{id}(t) \right] \end{aligned} \qquad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \qquad (2)$$

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \quad \text{where} \quad \varphi = c_1 + c_2 > 4 \qquad (3)$$

$$w = \frac{w_{min} - w_{max}}{iter_{max}} \times t + w_{max} \qquad (4)$$

where K and w are the constriction factor and the inertia weight, respectively, which improve the searching performance and convergence; c_1 and c_2 are the acceleration factors of the population, reflecting the influence of the global best and personal best, respectively; $U(0,1)$ is a uniformly distributed number from the interval (0,1); i is the individual number of the particle in the predetermined population; and t is the current iteration number. v_{id} is limited by its maximum value, v_{idmax}, which is usually set to be the maximum dynamic range of the corresponding variable.

Since the standard PSO was essentially developed in continuous space without constraints handling capability, we propose an improved PSO that handles binary variables and constraints, applicable for universal problems. In order to prevent premature convergence due to the limited number of feasible points in the binary space, our most important contribution is to add a “mutation” operator and an “archive” vector so that the particles can be released from the local optima and self-initiate a new searching.

A. *Discrete Binary Variable Handling Method*

The discrete binary PSO interprets the particle velocity for discrete binary variables as the probability of changing one state to the other (1 or 0) [16] in the way that

$$u_{id} = \begin{cases} 1 & U(0,1) < S(v_{id}) \\ 0 & U(0,1) \geq S(v_{id}) \end{cases}, \quad S(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \qquad (5)$$

$S(v_{id})$ is a sigmoid limiting transformation to limit the velocity-based probability to the interval (0,1). Now the maximum allowable velocity v_{idmax} is interpreted as the limit of probability that u_{id} will be 1 or 0.

B. *Constraint Handling Scheme*

Four constraint handling methods are commonly applied to tackle the equality and inequality constraints imposed upon objective functions in computation: *preserving feasibility method*, *repair algorithm*, *rejecting approach* and *penalty function*. *Preserving feasibility method* always keeps the initial point and intermediate points during iteration in the feasible space by certain updating scheme, such as saturation masking for bounded variables and embedded equality handling in

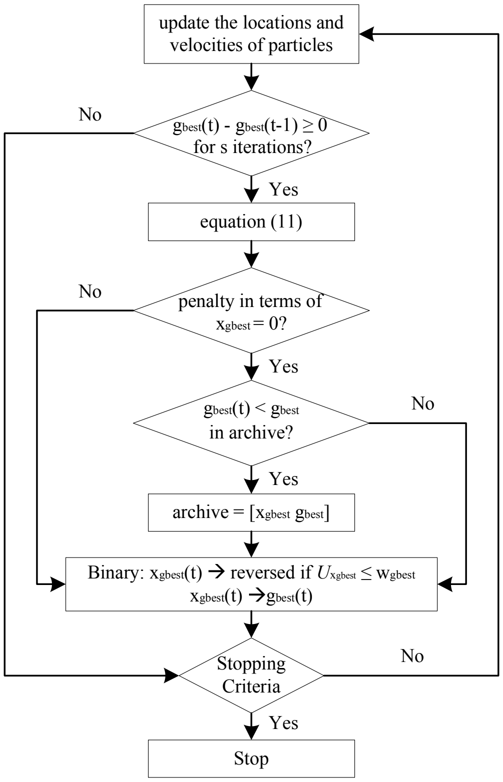
update the locations and velocities of particles
No
$g_{best}(t) - g_{best}(t-1) \geq 0$ for s iterations?
Yes
equation (11)
No
penalty in terms of $x_{gbest} = 0$?
Yes
$g_{best}(t) < g_{best}$ in archive?
No
Yes
archive = [x_{gbest} g_{best}]
Binary: $x_{gbest}(t)$ → reversed if $U_{xgbest} \leq w_{gbest}$
$x_{gbest}(t)$ → $g_{best}(t)$
Stopping Criteria
No
Yes
Stop

Figure 2. The data flow chart showing how the mutation operator and archive vector work

coding [8]. However, the variable updating scheme is highly problem-dependent, and especially suitable for solving the easy constraint expressions. When the constraints are nonlinear, or comprise polynomials and discrete expressions, the coding becomes challenging. *Repair algorithm* method is also problem specific and restoring feasibility may be as difficult as the optimization problem. *Rejecting approach* method evaluates every intermediate solution in all constraints and then rejects any solution violating the feasible search space. Hence, it consumes remarkable calculation time in complicated cases, moreover, as the number of variables increases, the heuristic computation mechanism needs to be largely improved to generate feasible solutions [17].

In contrast, *penalty function* incorporates constraints into the objective function through certain designed weights, so that the violation of constraints can be straightforwardly reflected as an unacceptably large value of the objective function [18]. In addition, the degree of constraint violation can be numerically expressed and analyzed through heuristic optimization process, other than being treated only as "feasible" or "infeasible" by the other constraint handling schemes. For a general constrained minimization problem in (6), its corresponding unconstrained objective function with penalties is shown in (7). Equation (8) and (9) provide an efficient form of penalty terms.

$$\min f(x) \qquad \text{subject to}$$

$$\begin{aligned} &g_j(x) \leq 0, j = 1,2,\cdots,n \\ &h_j(x) = 0, j = n+1, n+2,\cdots,m \end{aligned} \tag{6}$$

$$F(x) = f(x) + \sum_{j=1}^{m} k_j \times H_j(x) \tag{7}$$

$$\begin{aligned} D_j(x) &= max\{0, g_j(x)\}, \qquad j = 1,2,\cdots,n \\ D_j(x) &= max\{0, |h_j(x)|\}, \qquad j = n+1, n+2,\cdots,m \end{aligned} \tag{8}$$

$$k_j = (C\cdot t)^{\alpha}, \qquad H_j(x) = \sum_{j=1}^{m} D_j^{\beta}(x) \tag{9}$$

where $x = [x_1, x_2, \cdots, x_p]$ is the vector with p input variables; $f(x)$ is the original objective function subjected to n equality and $(m\text{-}n)$ inequality constraints; k_j and $H_j(x)$ are the penalty weight and the penalty factor of each constraint, respectively; C is a constant denoting the initial penalty effect; α and β are the constants defining the function form.

$$\begin{cases} P(\alpha,\beta):\quad P(1,1), P(1,2), P(2,2) \\ \text{Multi-stage:} \qquad k_i = \sqrt{t} \quad or \quad t\sqrt{t}, \quad H_i = \sum_{i=1}^{m} \theta_i \cdot D_i^{\gamma_i} \end{cases}$$

$$\text{where } \theta_i = \begin{cases} 10 \quad D_i \in [0, 0.001) \\ 20 \quad D_i \in [0.001, 0.1] \\ 100 \quad D_i \in (0.1, 1] \\ 300 \quad D_i \in (1, \infty) \end{cases}, \quad \gamma_i = \begin{cases} 1 \; D_i \in [0,1) \\ 2 \quad or\,else \end{cases} \tag{10}$$

Several dynamically-modified forms of penalty function have been developed [18][21] to address the most noticeable disadvantages of penalty function that high penalty weight results in easily getting trapped to the local optimum while low penalty weight may lead to non-convergence of the function. Since there is no clear conclusion on which penalty function outperforms the others in every problem, four forms of penalty functions in (10) are coded with the algorithm in our work for users' own test on different problems.

C. The Mutation Operator and Archive Vector

A fixed probability, w_1, of mutation for all design variables is introduced to enhance the global exploration capability of particles. If g_{best} fails to improve after s iterations, the velocities of certain real variables will be set to a random value in their corresponding velocity bounds; and the states of certain binary variables will be reversed, i.e. "1" changes to "0" and "0" changes to "1" as shown in (11).

$$\begin{cases} \text{real:} \quad v_{pid}(t+1) = U(0,1)\cdot V_{pidmax} \quad U_{pi}(0,1) \leq w_1 \\ \text{binary:} \quad x_{pid}(t+1) \Rightarrow \text{reverse} \quad U_{pi}(0,1) \leq w_1 \end{cases} \tag{11}$$

where v_{pid} denotes particle i's velocity of design variable x_p; $U_{pi}(0,1)$ is a random number in (0,1) for the particle i of x_p.

At the same time, g_{best} is tested to see if it satisfies the constraints, in other words, to see if the penalty in terms of x_{gbest} is zero. If so, both x_{gbest} and g_{best} value are stored in an "archive" vector, and then g_{best} is subject to a mutation by reversing some of the binary variables in x_{gbest} for certain probability w_{gbest}, so that the trend of all particles flying to the premature converging point can be disturbed and a new searching behavior will be initiated. If not, only the mutation of

g_{best} happens. At each time when a local minimum is found and x_{gbest} appears to be qualified solution, the current g_{best} will be compared with the previously archived g_{best}, once this value is improved, the archive vector will be updated with the new x_{gbest} and g_{best} until the stopping criteria is reached. The data flow chart is shown in Fig. 2.

However, it should be pointed out that using archive vector decelerates the calculation speed of optimization to a certain degree as the sacrifice of obtaining better results, especially noticeable for large complicated problems. Limiting the maximum number of times allowing the update of archive vector is able to efficiently balance the simulation speed and result quality. From our experiment, the number from 5 to 10 is the reasonable choice for problems with 16 to 20 design variables.

IV. Problem Formulation

We consider the unit commitment problem – on/off status, and how much power to produce from each generator – for an electric ship power system in the context of a set of defined mission segments to demonstrate our simulation-component-based optimal system design approach. For brevity, we only consider three candidate system designs, all configured in the same general architecture, i.e. a ring bus, and evaluated for the same single design objective – fuel economy.

It should be noted that the number and the choices of system design candidates do not affect the demonstration of our system co-optimization approach because even though the selection of equipment and configuration is involved in the co-optimization procedure, they are not technically handled by any optimization methodology. The created number of feasible design candidates is totally problem-dependent (what are the basic design requirements to choose the equipment? e.g. the minimum power capacity of the ship power system, the minimum/maximum number of gensets required/allowed onboard, and the total maximum volume/weight of gensets allowed onboard etc.) and software-dependent (how full is the model library?). Therefore, this paper is not focusing on how to create the design candidates because only its theoretical definition in the co-optimization procedure is essential for the users to solve general problems, but is focusing on the demonstration of the improved optimization algorithm because it is the key and the only shared information to implement co-optimization for every optimal system design problem.

The design candidates are generated to meet three basic design requirements: the total demanded power capacity of the equipped gensets, W_{ship}, the minimum required number of gensets, N_{min}, and the maximum allowed number of gensets, N_{max}. The optimal system is the one minimizing the total fuel cost by the total gensets installed onboard during the entire mission, i.e. predefined schedule of ship speed and equipment loads, while satisfying several control constraints, including the system real and reactive power balance as affected by the limited adjusting range of generator power factor, the generation capacity of each unit, and the requirement for system redundancy (to account for unplanned outages of any generator).

The real design variables are active power output of each generator, P, and power factor, pf. The binary variables are generator status, u, and generator reactive power status, v. We use u=1 for "ON", u=0 for "OFF", v=1 for lagging pf and v=0 for leading pf. n represents the number of gensets. Q and S denote reactive power and apparent power, respectively.

Therefore, the cost function of our system design can be represented as in (12).

$$\min f\left(P_g, pf, u, v\right) = \sum_{n=1}^{N} u_n \times \left\{ v_n \cdot f_{tn}\left[f_{gn}(P_{gn})\right] + \left(1 - v_n\right) \cdot f_{tn}\left[f_{gn}(P_{gn})\right] \right\} \tag{12}$$

where

$$P_{gn} = P_{goutn} + P_{exn} \tag{13}$$

$$P_{loadn} = P_{goutn} - P_{conn}, \quad Q_{loadn} = Q_{goutn} - Q_{conn} \tag{14}$$

$$Q_{goutn} = \frac{P_{goutn} \cdot \sqrt{1 - pf_n^2}}{pf_n} \tag{15}$$

$$P_{exn} = \alpha_n \cdot \left|Q_{goutn}\right|, \quad P_{conn} = \beta_n \cdot S_{goutn}, \quad Q_{conn} = \gamma_n \cdot S_{goutn} \tag{16}$$

where P_{gn} is the total active power needed by each generator, consisting of P_{goutn}, the active power output of generator and P_{exn}, the power consumption in the excitation circuit, which is proportional to the generator reactive power output Q_{goutn} by the factor α_n; f_{gn} is the function denoting the power efficiency of generator from the input to output; f_{tn} is the function denoting the hourly fuel consumption (l/h) of gas turbine in terms of the output power at shaft. f_{gn} and f_{tn} are both loaded through the user interface as user-defined curves. And their approximate equations can be obtained through data-fitting process and in the form of third-order polynomial function in (17) and (18), respectively. P_{loadn} and Q_{loadn} are the active power and reactive power contributed to load by each generator. P_{conn} and Q_{conn} whose values are proportional to the apparent power output of generator by the factor β_n and γ_n, respectively, are the active power loss and reactive power loss in the power converter modules. The unit of all power variables is in mega level.

$$f_{gn}\left(x\right) = a_{gn}x^3 + b_{gn}x^2 + c_{gn}x + d_{gn} \tag{17}$$

$$f_{tn}\left(x\right) = a_{tn}x^3 + b_{tn}x^2 + c_{tn}x + d_{tn} \tag{18}$$

Four constraints are considered in our problem.

A. Real and Reactive Power Balance Constraint

$$\sum_{n=1}^{4} u_n \cdot P_{loadn} - P_{load} = 0, \quad \sum_{n=1}^{4} u_n \cdot Q_{gn} - Q_{load} = 0 \tag{19}$$

where P_{load} and Q_{load} are the active power and reactive power demanded from the generator by the aggregated load at some point in a mission.

B. Generation Capacity Constraint Of Each Genset

$$u_n \cdot \sqrt{P_{gn}^2 + Q_{gn}^2} - P_{n,rate} \le 0 \quad n = 1,2,3,4 \tag{20}$$

where $P_{n,,rate}$ is the rated generation capacity of each synchronous generator.

C. Constraint Of Power Factor Adjusting Range

The generators are able to work in either lagging or leading power factor status and pf value is limited to certain intervals for stability purposes, so pf is a discontinuous real variable and the binary v_n is used to indicate its working status.

For leading power factor, pf is negative in value to indicate reactive power flow direction, and is bounded in [-1, $pf_{lead,max}$].

$$\begin{aligned} u_n \cdot (1 - v_n)(-1 - pf_n) &\le 0 \\ u_n \cdot (1 - v_n)(pf_{leadn,max} + pf_n) &\le 0 \end{aligned} \tag{21}$$

For lagging power factor, pf is positive and is bounded in [$pf_{lag,min}$, 1]:

$$\begin{aligned} u_n \cdot v_n \cdot (pf_{lagn,min} - pf_n) &\le 0 \\ u_n \cdot v_n \cdot (pf_n - 1) &\le 0 \end{aligned} \tag{22}$$

v_n ensures the unique sign of power factor by invalidating either (21) or (22) at every moment; and pf_n=-1 is the same meaning as pf_n=1, generating pure active power.

D. System Redundancy Constraint

Redundancy is important to reliability when designing an isolated power system in order to avoid single-point failures. U is the least number of generators that must be in service at any moment in a mission.

$$U - \sum_{n=1}^{4} u_n \le 0 \tag{23}$$

V. Simulation Results And Analysis

Gensets of four power ratings are available to be used in the candidate systems for this example: 25 MW, 30.3 MW, 44.7 MW and 50 MW. The total ship power demand is W_{ship}=150 MW, and the number of generators can be between N_{min}=4 and N_{max}=5. This yield three feasible combinations of gensets competing for the optimal design: [50 MW 50 MW 25 MW 25 MW], [50 MW 25 MW 25 MW 25 MW 25 MW] and [44.7 MW 44.7 MW 30.3 MW 30.3 MW], called candidate 1, candidate 2 and candidate 3, respectively.

The design parameters are shown in TABLE II. The parameters of the improved PSO are set to work for general cases as follows [12]: the population size is 100, the maximum iteration number is 10000, w_{max}=0.9, w_{min}=0.4, K=0.736, c_1=c_2=2.05, w_1=0.5 and w_{gbest}=0.2. The "archive" vector is set to update no more than 10 times for the acceptable simulation speed.

TABLE II. Design parameters of the electric ship power system

Parameter		Value
$[\alpha\ \beta\ \gamma]$		[0.1 0.001 0.001]
$pf_{lead,\max}$		-0.95
$pf_{lag,\min}$		0.5
U		2
50 MW	$[a_g\ \ b_g\ \ c_g\ \ d_g]$	[0.0005 -0.0308 1.4886 0.5984]
	$[a_t\ \ b_t\ \ c_t\ \ d_t]$	[0.0007 -0.0645 3.8207 3.2074]
44.7 MW	$[a_g\ \ b_g\ \ c_g\ \ d_g]$	[0.0008 -0.0415 1.5826 0.6047]
	$[a_t\ \ b_t\ \ c_t\ \ d_t]$	[0.0010 -0.0750 3.8664 2.6796]
30.3 MW	$[a_g\ \ b_g\ \ c_g\ \ d_g]$	[0.0015 -0.0521 1.4722 0.5876]
	$[a_t\ \ b_t\ \ c_t\ \ d_t]$	[0.0022 -0.1136 3.8783 2.4112]
25 MW	$[a_g\ \ b_g\ \ c_g\ \ d_g]$	[0.0019 -0.0545 1.4233 0.6179]
	$[a_t\ \ b_t\ \ c_t\ \ d_t]$	[0.0027 -0.1216 3.7577 2.6465]

TABLE III. Inputs to optimizer for the example system

Power Interval (MVA)	Selected Ship Speed (knots)	Apparent Power Demand (MVA)	Power Factor
[0 25]	8.52	17.07	0.895
[25 50]	20.87	42.94	0.813
[50 75]	23.3	55.42	0.800
[75 100]	29.08	89.89	0.899
[100 125]	31.95	114.09	0.938
[125 150]	34.64	139.10	0.989

First, we choose candidate 1 to just demonstrate the efficacy of the improved PSO by comparing the fuel consumption estimates for the system provided by three different methods – the improved PSO, the un-optimized default control (equal proportional allocations of power among generators), and a hybrid PSO-GA. With candidate 1 there are six possible power intervals corresponding to the combination of the two 25 MW and two 50 MW gensets as shown in the first column of TABLE III. We choose one arbitrary speed that falls within each of these six power intervals as the "mission" definition. The table shows the corresponding apparent power demand and power factor at load for each of speeds. These values are the inputs to the optimizer.

TABLE IV compares the simulation results for each of speeds derived by the improved PSO and the proportionally distributed power delivery method, which is applied by the generator controller block in VTB by default. It is obvious to see that the hourly fuel consumption of the gas turbine is considerably reduced by using the optimal control strategy instead of the default control method, by up to 52% at the test speeds, because the number and operation (i.e. active output power and power factor) of in-service gensets are optimally determined. In TABLE IV, the elements in the vectors are sequenced to denote the variables corresponding to the genset power level of [50 MW 50 MW 25 MW 25 MW]. To meet the requirement of system redundancy, 2 gensets are always in service at the first three test speeds even though one genset is capable of supplying sufficient power to the load.

It is noted that the biggest fuel saving comes from cutting off as many in-service gensets as possible while satisfying the

TABLE IV. Comparison of hourly fuel consumption obtained from the proportionally distributed power delivery method (PDPD) and the improved PSO

Ship Speed (knots)	Hourly Fuel Consumption (l/h)			Operation of the Gensets	Active Power Output of the Generators (MW)	Power Factor at the Generator Output
	PDPD Method	Improved PSO	Improvement (%)			
8.52	11181.6	5385.6	51.84	[0 0 1 1]	[0 0 15.45 0]	[1 1 0.893 1]
20.87	16261.2	11091.6	31.79	[1 0 1 0]	[35.35 0 0 0]	[0.812 1 1 1]
23.3	18324	14018.4	23.50	[1 0 1 0]	[29.61 0 15.02 0]	[0.662 1 1 1]
29.08	26326.8	24379.2	7.40	[1 1 1 0]	[30.82 37.13 12.86 0]	[0.616 1 1 1]
31.95	33926.4	32083.2	5.43	[1 1 1 1]	[36.60 39.97 14.94 16.00]	[1 0.985 0.665 0.699]
34.64	45788.4	45072	1.56	[1 1 1 1]	[46.78 47.32 22.46 22.44]	[0.965 1 1 0.923]

TABLE V. Comparison of optimization performance between a Hybrid PSO-GA [20] and the improved PSO

Ship Speed (knots)	Hourly Fuel Consumption (l/h)			Standard Deviation		
	Hybrid PSO-GA	Improved PSO	Fuel Saving by the Improved PSO (l/h)	Hybrid PSO-GA	Improved PSO	Improvement (%)
8.52	5385.6	5385.6	0	8.735	6.443	26.24
20.87	11091.6	11091.6	0	8.495	6.215	26.84
23.3	14370.3	14018.4	351.9	10.342	7.002	32.30
29.08	24544.3	24379.2	165.1	11.235	11.063	1.53
31.95	32321.7	32083.2	238.5	12.850	9.917	22.82
34.64	45142.3	45072	70.3	7.363	5.134	30.30

load power demand, as compared with just fine tuning the operation of gensets. Therefore, the advantage of using the optimal control strategy is becoming less notable while the load power is increasing, because the number of in-service gensets from using two different control strategies is becoming closer. Even so, there is still a considerable saving on the fuel cost in the long term from using the optimal system design, not to mention that the ship spends most of its time in the low-speed cruise mode, as indicated in the co-optimization problem later.

Using the same parameter setting, we also compare the performance of the improved PSO with another heuristic optimization algorithm [20], which is introduced as a hybrid PSO-GA method and has been proven to outperform the original binary version of PSO, in TABLE V. We run 200 trial simulations for each mission segment so that the accuracy and reliability of the two algorithms are accurately compared in terms of the minimum value of cost function and the standard deviation from the average of enough solutions. It is clear that the improved PSO is capable to consistently locate a better result than the hybrid PSO-GA. The optimal value is improved up to 2.4% in the test cases. In addition, the improved PSO consistently generates more reliable solutions than the hybrid PSO-GA, as indicated by less standard deviation in all test cases, up to 32%.

Secondly, we employ our validated PSO to demonstrate the mission-oriented co-optimization approach in terms of the three candidate designs. We apply the improved PSO on each of speed segments to obtain the corresponding best operating point in terms of fuel cost and the comparison of the three candidates at their best performance, as shown in Fig. 4. It is clear that candidate 1 and candidate 2 consume the equal amount of fuel at low speed because the genset operation is the same, but candidate 2 starts to consume more fuel at speeds above 26 knots because it has to always involve one more genset to provide enough load power. It is difficult to tell whether candidate 1 or candidate 3 is better because both consume the least amount of fuel at certain speed. However, for a given mission profile, which in our study is the actual speed schedule measure for a 24-hour period for DDG-51 [19] as shown in Fig. 5, the optimal design, which is candidate 1, can be uniquely decided from the three candidates by their total fuel cost, as shown in TABLE VI. It must be noted that our simulation-component-based co-optimization approach can be applied to any type of system designs with mission considered. The generation of design candidates is based on the user's expertise and the simulation software tool, but the improved PSO developed for control strategy optimization is applicable to any types of functions.

TABLE VI. Comparison of the total fuel consumption of the three candidates optimized by the improved PSO during the whole mission

Candidate	Total Fuel Consumption (l)
Candidate 1: [50 50 25 25] MW	201698.1680
Candidate 2: [50 25 25 25 25] MW	202670.4230
Candidate 3: [44.7 44.7 30.3 30.3] MW	209158.5273

VI. CONCLUSION

In this paper, we introduce an improved PSO for solving mission-oriented co-optimization of system designs. The improved PSO employs a binary handling method and dynamic penalty function to tackle universal constrained cost functions for real system design more efficiently than the other existing optimization algorithms. Moreover, we add a "mutation" operator and an "archive" vector into the global exploration process of PSO so that the premature convergence due to the limited status of binary variables can be avoided to a large extent. We integrate this optimizer with a toolbox that improves the flexibility of simulation-component-based approach by automatically generating compatible simulation formats corresponding to the user-requested analyses in its associated simulation software tool. Designing the optimal

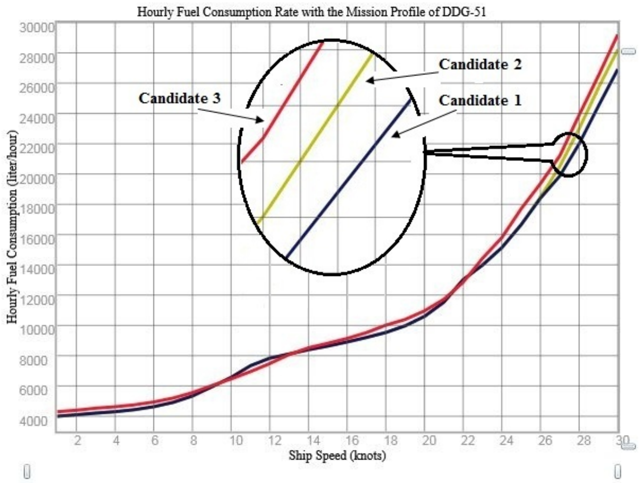

Figure 3. Comparison of hourly fuel consumption of the three candidates using the improved PSO

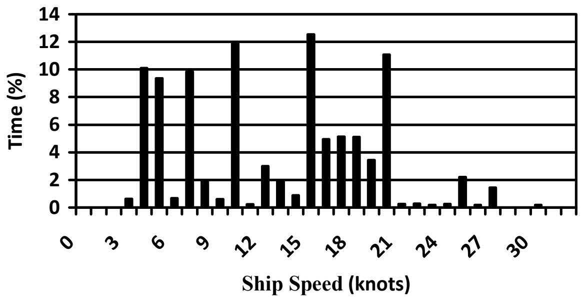

Figure 4. The mission profile used in the co-optimization example

electric ship power system for a mission-based unit commitment problem is considered to demonstrate our simulation-component-based co-optimization approach and the efficacy of the improved PSO algorithm. Three design candidates are generated and one is employed to validate the improved PSO algorithm. The simulation results indicate that the improved PSO outperforms another hybrid PSO-GA method by consistently generating more accurate global optimum and more reliable solutions. The accuracy and the reliability of solutions are improved by up to 2.4% and 32%, respectively, in our test cases. Then the co-optimization approach for system designs is realized by applying the improved PSO to the control strategy optimization of each candidate and comparing their best performance during the same missions. It is clear that our proposed system design methods can be easily applied to general co-optimization problems and assist the system designers to efficiently involve equipment selection, configuration selection and the system operating mission in the early stage design.

Our future work will apply this methodology to the multi-objective optimization problem, e.g. the unit commitment problem of electric ship power system considering the optimal thermal cooling system design.

REFERENCES

[1] T. Moore, "HEV control strategy: implications of performance criteria, system configuration and design, and component selection," Proc. American Control Conference, vol. 1, pp.679-683, June 1997.

[2] S. Akella, N. Sivashankar, and S. Gopalswamy, "Model-based systems analysis of a hybrid fuel cell vehicle configuration," Proc. American Control Conference, vol. 3, pp.1777-1782, June 2001.

[3] Y. Zhang, and K. Lum, "Integrated-optimal design of airplane and flight control using genetic algorithms," IEEE CEC 2007, Singapore, pp. 2980-2987, Sept. 2007.

[4] N. Hanini, B. Tabbache, A. Kheloui and T. Roubache, "Sizing methodology of EV drive system based on optimal power efficiency," 2008 International SPEEDAM, Ischia, pp. 1043-1048, Jun. 2008.

[5] S. Amari, and G. Dill, "Redundancy optimization problem with warm-standby redundancy," Proc. 2010 Annual RAMS, pp. 1-6, Jan. 2010.

[6] J. Verdaasdonk, H. Grimmelius, C. Bil, and S. Anavatti, "Comprehensive design tool for sizing and simulation of autonomous underwater vehicles," OCEANS 2007 – Europe, pp. 1-5, June 2007.

[7] N. Barsoum, and P. Vacent, "Balancing cost, operation and performance in integrated hydrogen hybrid energy system," 1st Asia International Conference on Modeling and Simulation, pp. 14-18, Mar. 2007.

[8] M. AlRashidi, M. El-Hawary, "Hybrid Particle Swarm Optimization approach for solving the discrete OPF problem considering the valve loading effects," IEEE Trans. on Power Systems, vol. 22, no. 4, pp. 2030-2038, Nov. 2007.

[9] P. Venkatesh, R. Gnanadass and N. Padhy, "Comparison and application of evolutionary programming techniques to combined economic emission dispatch with line flow constraints," IEEE Trans. on Power Systems, vol. 18, no. 2, pp. 688-697, May 2003.

[10] K. Wong and Y. Wong, "Genetic and genetic/simulated-annealing approaches to economic dispatch," IEEE Proc. Gener. Trans. Distrib., vol. 141, issue 5, pp. 507-513, Sep. 1994.

[11] S. Affijulla and S. Chauhan, "A new intelligence solution for power system economic load dispatch," 10th International Conference on Environment and Electrical Engineering, , pp. 1-5, May 2011.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," IEEE Proc. the International Conference on Neural Networks, Perth, Australia, pp. 1942-1948, 1995.

[13] M. Abido, "Optimal design of power-system stabilizers using Particle Swarm Optimization," IEEE Trans. on Energy Conversion, vol. 17, no. 3, pp.406-413, Sep. 2002.

[14] http://vtb.engr.sc.edu/vtbwebsite/#/Overview,Virtual Test Bed Overview

[15] G. Conklin, R. Dougal, and B. Langland, "Using simulation generation templates to interface simulation analysis tools with design space models," unpublished.

[16] L. Wang and C. Singh, "Unit commitment considering generator outages through a mixed-integer particle swarm optimization algorithm," Applied Soft Computing, vol. 9, pp.947-953, 2009.

[17] J. Joines, C. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's," IEEE Int. Conf. Evol. Comp, pp.579-585, Jun. 1994.

[18] K. Parsopoulos, M. Vrahatis, "Particle Swarm Optimization method for constrained optimization problems," Intelligent Technologies - Theory and Applications: New Trends in Intelligent Technologies, pp. 214–220. IOS Press, 2002.

[19] S. Vijlee, A. Ouroua, L. Domaschk, and J. Beno, "Directly-coupled gas turbine permanent magnet generator sets for prime power generation on board electric ships," 2007 Electric Ship Technologies Symposium, pp. 340-347, May 2007.

[20] R. R. Tan, "Hybrid evolutionary computation for the development of pollution prevention and control strategies," Journal of Cleaner Production 15, pp. 902-906, 2007

[21] J. Joines, C. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's," IEEE Int. Conf. Evol. Comp, pp.579-585, Jun. 1994.