MDPI

Article

Adaptive multi-level search for global optimization: An integrated swarm intelligence-metamodelling technique

Guirong Dong [1,‡], Chengyang Liu [2,‡], Dianzi Liu [2,*] and Xiaoan Mao [3]

[1] Faculty of printing, packaging engineering and digital media technology, Xi'an University of Technology, Xi'an, China
[2] School of Engineering, Faculty of Science, University of East Anglia, Norwich, UK; e-mail@e-mail.com
[3] Faculty of Engineering, University of Leeds, Leeds LS2 9JT, UK
* Correspondence: dianzi.liu@uea.ac.uk
‡ These authors contributed equally to this work.

Abstract: Over the last decade, metaheuristic algorithms have emerged as a powerful paradigm for global optimization of multimodal functions formulated by nonlinear problems arising from various engineering subjects. However, numerical analyses of many complex engineering design problems may be performed using finite element method (FEM) or computational fluid dynamics (CFD), by which function evaluations of population-based algorithms are repetitively computed for seeking a global optimum. It is noted that these simulations become computationally prohibitive for design optimization of complex structures. To efficiently and effectively address this class of problems, an adaptively integrated swarm intelligence-metamodelling (ASIM) technique enabling multi-level search and model management for the optimal solution is proposed in this paper. The developed technique comprises two steps: in the first step, a global-level exploration for near optimal solution is performed by adaptive swarm-intelligence algorithm, and in the second step a local-level exploitation for the fine optimal solution is studied on adaptive metamodels, which are constructed by multipoint approximation method (MAM). To demonstrate the superiority of the proposed technique over other methods, such as conventional MAM, particle swarm optimization, hybrid cuckoo search, water cycle algorithm in terms of computational expense associated with solving complex optimization problems, one benchmark mathematical example and two real-world complex design problems are examined. In particular, the key factors responsible for the balance between exploration and exploitation are discussed as well.

Keywords: adaptive multi-level search; metamodel-based hybrid algorithm; particle swarm optimization; multipoint approximation method

Citation: Dong, G.; Liu, C.; Liu, D., Mao, X. ASIM. *Appl. Sci.* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

1. Introduction

With tremendous advances in computational sciences, information technology and artificial intelligence, design optimization becomes increasingly popular in many engineering subjects, such as mechanical, civil, structural, aerospace, automotive and energy engineering. It helps shorten the design-cycle time and identify creative designs that are not only feasible, but also progressively optimal, given predetermined design criteria.

At the outset of design optimization, running a gradient-based algorithm with a multi-start process proves to be very successful in finding global optimum of simple problems when gradient information is available [1]. While under the pressure of being faced with increasingly complex optimization problems in which derivative information is unreliable or unavailable, researchers gradually focus on the development of derivative-free optimization methods [2] and metaheuristic methods to address this issue. Followed by Glover's convention [3], modern metaheuristic algorithms such as simulated annealing(SA)[4], genetic algorithms(GA) [5,6], particle swarm optimization(PSO) [7] and ant colony optimization(ACO) [8] have been applied with good success

in solving complex nonlinear optimization problems [9,10]. The popularity of these nature-inspired algorithms lies in their ease of implementation and the capability to obtain the solution close to the global optimum. However, for many real-life design problems, more than thousands of calls for high-fidelity simulations (for example, computational fluid dynamics simulation) may be executed to seek a near-optimal solution. This is the overwhelming part of the total run time required in the design cycle. Thus, it is desirable to retain the appeal of metaheuristic algorithms on global searching while replacing as many as possible calls to the solver with evaluations on metamodels for the purpose of less computational cost [11].

The typical techniques for metamodel building include Kriging [12], polynomial response surface (PRS) [13], radial basis function (RBF) [14], artificial network (ANN) [15], etc. Among them, PRS and ANN are regression methods that have advantages in dealing with convex problems; Kriging and RBF belong to interpolation methods that are more appropriate for non-convex or multi-modal problems [16]. Therefore, metamodels have been successfully employed to assist evolutionary optimizations [17–19] and PSO method. For example, Tang et al.[20] proposed a hybrid surrogate model formed from a quadratic polynomial and a RBF model to develop a surrogate-based PSO method and applied it to solve mostly low-dimensional test problems and engineering design problems. Regis [21] used RBF surrogates on PSO to identify the most promising trail position surrounding the current overall global best position for solving a 36-dimensional bioremediation problem. However, inherent nature of PSO method leads to extremely large number of calls for function evaluations, which might be prohibitive in simulation-based optimization.

In this paper, an adaptively integrated swarm intelligence-metamodelling technique (ASIM) is proposed, which combines the multi-level search and model management during the entire optimization process. It orients the solution of the approximate model to the global optimum with a smaller number of iterations of analyses and achieves a higher level of efficiency than conventional approximation methods. Meanwhile, the model management in the optimization process has been established, which integrates an adaptive trust-region strategy with a space reduction scheme implemented in the multipoint approximation method (MAM) framework. The model management has been able to facilitate the optimization process and improve the robustness during iterations. Especially, it has allowed a small perturbation to be assigned to the current position in case of no update of the optimum position. The developed ASIM makes full use of global-exploration potential of PSO and local-exploitation advantage of MAM to efficiently and accurately seek the global optimal solution with low computational cost. By comparison with the results by other algorithms such as conventional MAM, particle swarm optimization [22], hybrid cuckoo search [23], water cycle algorithm [24], etc., the superiority of ASIM has been demonstrated in terms of computational expense and accuracy throughout three case studies.

2. Brief review of multipoint approximation method (MAM)

The MAM [25,26] was proposed to tackle black-box optimization problem and has gained continuous development in recent years, e.g. Polynkin [27] enhanced MAM to solve large scale optimization problems, one of which is the optimization of transonic axial compressor rotor blades, Liu [28] implemented discrete capability into MAM. Recently, Caloni [29] has applied MAM to solve a multi-objective problem. Based on response surface methodology, multipoint approximation method (MAM) aims at constructing mid-range approximations and is suitable to solve complex optimization problems owing to: 1) producing better quality of approximations that are sufficiently accurate in a current trust region and, 2) the affordability in terms of computational costs required for their building. These approximation functions have a relatively small number ($N + 1$ where N is number of design variables) of regression coefficients to be determined and the corresponding least squares problem can be solved easily [25].

In general, an black-box optimization problem can be formulated as

$$\begin{aligned} &min\ f(\boldsymbol{x}) \\ &s.t.\ g_j(\boldsymbol{x}) \leq 1\ (j = 1, ..., M) \\ &\quad A_i \leq x_i \leq B_i\ (i = 1, ..., N) \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ refers to the vector of design variables; A_i and B_i are the given lower and upper bounds of the design variable x_i ; N is the total number of the design variables; $f(\boldsymbol{x})$ is the objective function; $g_j(\boldsymbol{x})$ is the j-th constraint function and M is the total number of the constraint functions.

In order to represent the detailed physical model using the response functions and reduce the number of calls for the response function evaluations, the MAM replaces the optimization problem with a sequence of approximate optimization problems as follows:

$$\begin{aligned} &min\ \tilde{f}^k(\boldsymbol{x}) \\ &s.t.\ \tilde{g}_j^k(\boldsymbol{x}) \leq 1\ (j = 1, ..., M) \\ &\quad A_i \leq A_i^k \leq x_i \leq B_i^k \leq B_i\ (i = 1, ..., N) \end{aligned} \tag{2}$$

where $\tilde{f}^k(\boldsymbol{x})$ and $\tilde{g}_j^k(\boldsymbol{x})$ are the functions which approximate the functions $f(\boldsymbol{x})$ and $g_j(\boldsymbol{x})$ defined in Equation 1; A_i^k and B_i^k are the side constraints of a trust sub-region; and k is the iteration number.

Comparing with the time spent by the evaluation of the actual response functions $g_j(\boldsymbol{x})$, the selected form of approximate functions $\tilde{g}_j^k(\boldsymbol{x})$ $(j = 0, ...M)$ remarkably reduces the computational expense and adequately improves the accuracy in a current trust region. This is achieved by appropriate planning of numerical experiments and use of the trust region defined by the side constraints A_i^k and B_i^k. Once the current sub-optimization problem is solved, the sub-optimal solution becomes the starting point for the next step. Meanwhile, the move limits are modified and the trust region is resized [25,26]. Based on these information, the metamodel is updated in the next iteration until eventually the optimum is reached.

The process of metamodel building in MAM can be described as an assembly of multiple surrogates into one single metamodel using linear regression. Therefore, there are two stages of metamodel building.

In the first stage, the parameter $\boldsymbol{a}_l$ of an individual surrogate φ_l is determined by solving a weighted least squares problem involving n fitting points as

$$min \quad \sum_{i=1}^{n} \omega_i [F(\boldsymbol{x}_i) - \varphi_l(\boldsymbol{x}_i, \boldsymbol{a}_l)]^2 \tag{3}$$

where ω_i denote the weighting parameters and F is the original function needs to be approximated. Here, the selection of weighting factors ω_i should reflect the quality of the objective function and the location of a design point with respect to the border between the feasible and the infeasible design subspace [30], which are defined as

$$w_i = w_i^o \cdot w_i^c \tag{4}$$

$$w_i^o = [\frac{f(\boldsymbol{x}^k)}{f(\boldsymbol{x}_i)}]^\beta \tag{5}$$

$$w_i^c = \begin{cases} 1 & \text{for objective } f(x) \\ [g(\boldsymbol{x}) + 1]^\alpha & \text{if } g(\boldsymbol{x}) \leq 0 \\ [g(\boldsymbol{x}) + 1]^{-\alpha} & \text{if } g(\boldsymbol{x}) \geq 0 \end{cases} \tag{6}$$

where $\alpha, \beta > 0$ are user defined constants, here $\alpha = 4, \beta = 1.5$ are used; $\boldsymbol{x}^k$ is the starting point in k_{th} iteration and $\boldsymbol{x}_i$ is the i_{th} design point in the fitting points. With this definition,

a point with a larger objective function has a smaller weighting coefficient component w_i^o. For a constraint function $g(\boldsymbol{x})$, a point which is much closer to the boundary of the feasible region of $g(\boldsymbol{x})$, is given a larger weighting coefficient component w_i^c. For building a surrogate of the objective function $f(\boldsymbol{x})$, the weighting coefficient w_i will only consider the component w_i^o. But for building a surrogate of the constraint function $g(\boldsymbol{x})$, the weighting coefficient w_i will also take the constraint component w_i^c into consideration.

It should be noted here that in MAM, both the objective and constraint functions will be approximated by Equation 3. The simplest case of φ_l is the first order polynomial metamodel and more complex ones are intrinsically linear functions (ILF) that have been successfully applied for solving various design optimization problem [25,28,29]. ILF are nonlinear but they can be led to linear ones by simple transformations. Currently, five functions are considered in the regressor pool $\{\varphi_l(\boldsymbol{x})\}$ as

$$\begin{aligned}
\varphi_1(\boldsymbol{x}) &= a_0 + \sum_{i=1}^{N} a_i x_i \\
\varphi_2(\boldsymbol{x}) &= a_0 + \sum_{i=1}^{N} a_i x_i^2 \\
\varphi_3(\boldsymbol{x}) &= a_0 + \sum_{i=1}^{N} a_i / x_i \\
\varphi_4(\boldsymbol{x}) &= a_0 + \sum_{i=1}^{N} a_i / x_i^2 \\
\varphi_5(\boldsymbol{x}) &= a_0 \prod_{i=1}^{N} x_i^{a_i}
\end{aligned} \tag{7}$$

In the second stage, for each function ($f(\boldsymbol{x})$ or $g(\boldsymbol{x})$), different surrogates are assembled into one metamodel as

$$\tilde{F}(\boldsymbol{x}) = \sum_{l=1}^{n_l} b_l \varphi_l(\boldsymbol{x}) \tag{8}$$

where n_l is the number of surrogates applied in the model bank $\{\varphi_l(\boldsymbol{x})\}$, and b_l is the regression coefficient corresponding to each surrogate $\varphi_l(\boldsymbol{x})$, which reflects the quality of the individual $\varphi_l(\boldsymbol{x})$ on the set of validation points. Similar to Equation 3, b_l can be determined in the same manner as

$$min \quad \sum_{i=1}^{n} \omega_i \left[F(\boldsymbol{x}_i) - \tilde{F}(\boldsymbol{x}_i, \boldsymbol{b})\right]^2 \tag{9}$$

It should be noted that in the process of metamodel building, the DOE is fixed, i.e., ω_i remains unchanged across the aforementioned stages.

The Figure 1 illustrates the main steps of in MAM. Note that once the metamodels for the objective and constraint functions have been built, the constrained optimization subproblem formulated in the trust region (Equation 2) could be solved by any existing optimizers. In this paper, the sequential quadratic programming (SQP) method [31] is applied to solve the constrained optimization subproblem for the optimal solution. Since numerical optimization solvers like SQP are deterministic, the quality of the obtained solution is highly sensitive to the initial point. In other words, MAM could not perform the global search very well. To address this issue, ASIM framework in Section 4 has been proposed to integrate the stochastic nature with the exploratory search ability of PSO for the global optimal solution.

3. Brief review of particle swarm optimization (PSO)

Particle swarm optimization (PSO), inspired from swarm behaviors in nature such as fish and bird schooling, was developed by Kennedy and Eberhart [32]. Since then,

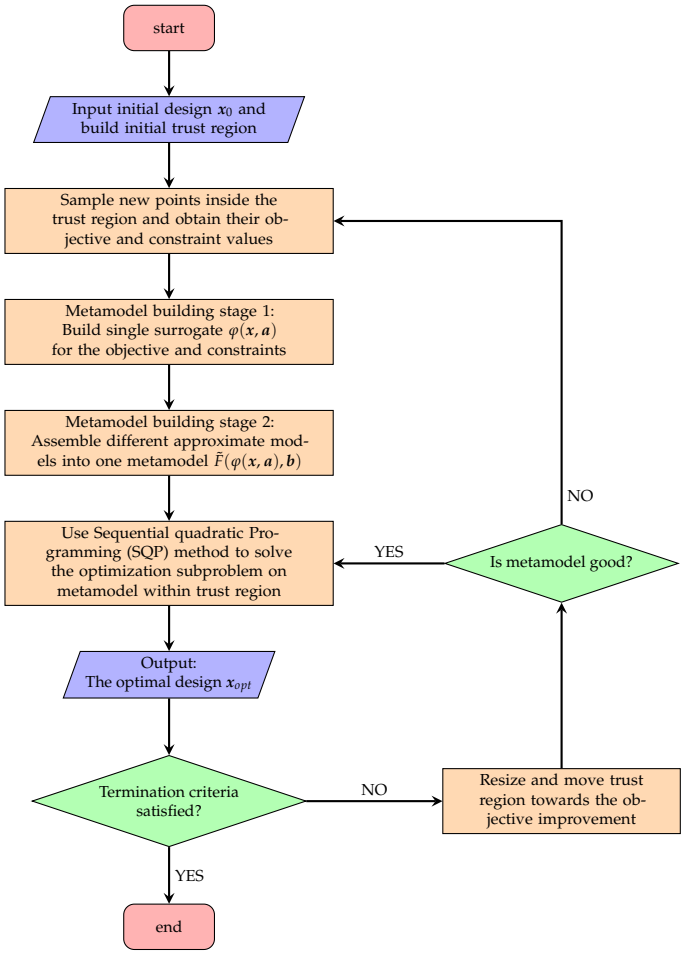

Figure 1. Flow chart of MAM

PSO has attracted a lot of attention and been developed as a main representative form of swarm intelligence. PSO has been applied to many areas, such as image and video analysis applications, engineering designs and scheduling applications, classification and data mining, etc [33]. There are at least twenty PSO variants, as well as hybrid algorithms obtained by combining PSO with other existing algorithms, which are also becoming increasingly popular [34–36].

To integrate PSO with MAM to find the global optimum, adaptive multi-level search is proposed in this paper. PSO is employed for the global-level exploration in the first step. A number of particles are first placed in the search space of the optimization problem with initial positions and velocities. However, the particles can fly over the entire design space not only determined by the individual and collective knowledge of positions from the global-level search, but also based on the 'local' information of each particle. Here, the 'local' information means the local-level exploitation in the second step. In the neighborhood of each particle, an adaptive metamodel is constructed using MAM in Section 2, which replaces the original optimization problem by a sequence of mathematical approximations that use much simpler objective and constraint functions. Hence, the critical information about individual constraint functions is kept and this, leads to the improved accuracy of metamodels. During the process of metamodel building, each particle is endowed with the horizon in the surrounding region, and then is refined with the current individual position so as to boost the possibility of finding an optimal position. Eventually, the swarm as a whole, like a flock of birds collectively foraging for food while each bird is brilliant to directly find the most tasty food within the limited horizon, has ability to move toward to a global optimum.

Each particle in PSO represents a point in the design space of an optimization problem with an associated velocity vector. In each iteration of PSO, the velocity vector is updated by using a linear combination of three terms shown in Equation 10. The first term called inertia or momentum, reflects a memory of the previous flight direction and prevents the particle from changing direction thoroughly. The second term, called the cognitive component, describes the tendency of particles returning to the previously found best positions. The last term, called the social component, quantifies the group norm or standard that should be attained. In other words, each particle tends to move toward the position of the current global best $\boldsymbol{gbest}$ and the location of the individual best $\boldsymbol{pbest}$, while moving randomly [33]. The aim is to find the global best among all the current best solutions until the objective no longer improves or a certain number of iterations are reached. The standard iteration procedure of PSO is formulated as follows:

$$\begin{aligned} \boldsymbol{V}_i^{t+1} &= \omega \boldsymbol{V}_i^t + \alpha \boldsymbol{\epsilon}_1 (\boldsymbol{pbest}_i^t - \boldsymbol{x}_i^t) + \beta \boldsymbol{\epsilon_2} (\boldsymbol{gbest}_i^t - \boldsymbol{x}_i^t) \\ \boldsymbol{x}_i^{t+1} &= \boldsymbol{x}_i^t + \boldsymbol{V}_i^{t+1} \end{aligned} \tag{10}$$

where ω is the parameter called inertial weight, t is the current iteration number, α and β are parameters called acceleration coefficients, ϵ_1 and ϵ_2 are two homogeneously distributed random vectors generated within the interval $[0, 1)$, respectively. If the values of ω, ϵ_1 and ϵ_2 are properly chosen ($\epsilon = \alpha + \beta > 4$ and $\omega = \frac{2}{\epsilon - 2 + \sqrt{\epsilon^2 - 4\epsilon}}$), it has been proved that PSO could converge to an optimum [37].

Even PSO has been used in a variety of industry applications, it should be noted that the standard PSO suffers the disadvantages of information loss in the penalty function and highly computational cost, especially in solving constrained optimization problems. Therefore, the proposed ASIM framework in the following section takes the advantage of PSO in global searching and reduces the burden on computation by introducing the metamodel building technique, model management and trust region strategy.

4. Adaptively integrated swarm intelligence-metamodelling framework

4.1. Methodology of ASIM framework

In this paper, an adaptively integrated swarm intelligence-metamodelling (ASIM) framework has been proposed to perform the search for the optimal solution in two levels.

In the first level optimization, also known as exploration, a number of entities are initially placed in the search space of the particular optimization problem with respective positions $\boldsymbol{x}_i^t$ and velocities $\boldsymbol{v}_i^t$. Each particle i has its movement controlled by Equation 10. The final global best solution will be obtained only if the objective no longer improves or after a certain number of iterations. However, distinguished from the conventional PSO, each particle also gains the insight within its neighborhood. That forces each particle to refine the personal best position by exploiting its neighborhood, which is known as the second level optimization. In this local level search, an adaptive metamodel will be built by MAM within a trust region surrounding the particle, and then the personal best solution $\boldsymbol{x}_{i,MAM}$ obtained by MAM will be regarded as a local refinement in position. Following that, the personal and global best position $\boldsymbol{pbest}^t, \boldsymbol{gbest}^t$ will be determined and updated till the termination criterion is satisfied. To sum up, the surrogate helps guide the search direction of each particle and assists to refine the current overall best position until the final global best solution is found. Eventually, the swarm as a whole, moves close to a global optimum of the objective function. The flowchart of ASIM framework has been depicted in Figure 2.

It is worth noting that there are three rules applied to compare solutions during the optimization process:

1. Any feasible solution is preferred to any infeasible solution;
2. Among feasible solutions, the one with a better objective function value is preferred.

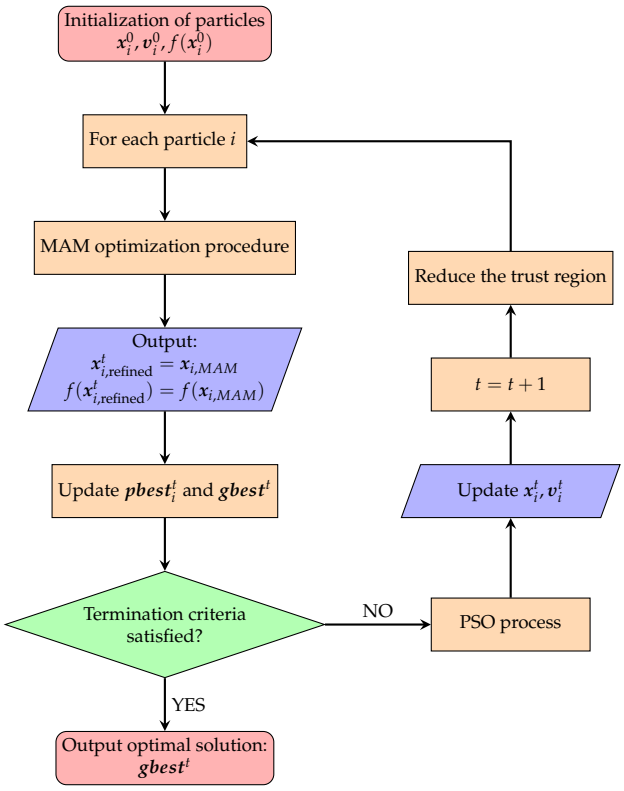

Figure 2. Flow chart of ASIM Framework

3. Among infeasible solutions, the one having a fitness value with smaller constraint violations is preferred. In the current implementation, the fitness function is defined by

$$Fitness(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \text{if } \boldsymbol{x} \text{ is feasible} \\ f(\boldsymbol{x}) * \prod[g_j(\boldsymbol{x})]^2 & \text{elseif } f(\boldsymbol{x}) \geq 0 \\ f(\boldsymbol{x}) + |f(\boldsymbol{x})| \cdot \prod[g_j(\boldsymbol{x}) - 1]^2 & \text{elseif } f(\boldsymbol{x}) < 0 \end{cases} \quad (11)$$

4.2. Model management

4.2.1. Strategy for particles 'flying out' in PSO

For particles located outside the boundary, they should adjust their positions according to the formulations determined by the current bounds as follows:

$$x_{i,k} = \begin{cases} a[k] + \gamma \cdot (b[k] - a[k]) & if\ x_{i,k} \leq a[k] \\ b[k] - \gamma \cdot (b[k] - a[k]) & if\ x_{i,k} \geq b[k] \end{cases} \quad (12)$$

where $x_{i,k}$ means the k^{th} dimensional position of $\boldsymbol{x}_i^t$, $a[k]$ and $b[k]$ are k^{th} dimensional side constraints, γ is a relatively small value randomly generated from the range $(0, 0.1)$. This perturbation of positions could actually force the particles back into the design space if particles violate the boundary constraints during the entire search process, and ensure the efficiency and accuracy in local exploitation.

4.2.2. Modified trust region strategy in MAM

The aim of the trust region strategy in MAM is to control the quality of a metamodel constructed. When the approximation gets better, the trust region will be further reduced for the optimal solution. The track of the trust regions also indicates a path of the direction from the initial starting point to the optimum over the entire searching domain.

At each iteration, a new trust region must be updated, i.e., its new size and its location have to be specified. Several indicators are formulated to support the control of the trust region and facilitate the search process. The basic knowledge about these indicators was also introduced in [38].

Table 1: Six indicators in MAM

1st indicator	The quality of metamodel approximation		
	Good	Reasonable	Bad
2nd indicator	Location of the sub-optimum point $\boldsymbol{x}^{k+1}$ with respect to trust region		
	Boundary	Internal	External
3rd indicator & 4th indicator	The angle between the last two move vectors		
	Backward $\left(\frac{\pi}{2} \leq \theta \leq \frac{3\pi}{2}\right)$ Parallel		Forward $\left(-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\right)$ Curved
5th indicator	Termination criterion: size of the current region		
	Small		Large
6th indicator	Value of the most active constraint		
	Close from the boundary		Far from the boundary

The first indicator is to evaluate the quality of the metamodel and focused on the accuracy of the constraint approximations at the obtained sub-optimal point $\boldsymbol{x}^{k+1}$. This is based on the following equation:

$$E^k = Max\left(\left|\frac{\tilde{g}\left(\boldsymbol{x}^{k+1}\right) - g\left(\boldsymbol{x}^{k+1}\right)}{g\left(\boldsymbol{x}^{k+1}\right)}\right|\right) \tag{13}$$

where $\tilde{g}\left(\boldsymbol{x}^{k+1}\right)$ and $g\left(\boldsymbol{x}^{k+1}\right)$ are normalized functions of the approximate and true constraints at the sub-optimal point $\boldsymbol{x}^{k+1}$, respectively. In this way, a single maximal error quantity between explicit approximation and implicit simulation is defined. Then, the quality of metamodel can be labeled as 'bad', 'reasonable' or 'good' shown below.

$$E^k \Rightarrow \begin{cases} \geq 0.25 \cdot S^k & \Rightarrow \text{'Bad'} \\ \leq 0.01 \cdot S^k & \Rightarrow \text{'Good'} \\ Else & \Rightarrow \text{'Reasonable'} \end{cases} \tag{14}$$

where S^k represents the maximum ratio of the dimension length between the present trust region and the entire design space is defined by

$$S^k = Max\left(\frac{B_i^k - A_i^k}{B_i - A_i}\right) \quad (i = 1, ..., d) \tag{15}$$

The second indicator is to indicate the location of the current iterate $\boldsymbol{x}^{k+1}$ in the present search subregion. For each dimension, if none of the current move limits $(\boldsymbol{A}^k, \boldsymbol{B}^k)$ is active, this solution is regarded as 'Internal', otherwise it is viewed as 'External'.

The third and fourth indicator reflects the movement history for the entire optimization process. For this purpose, the angle between the last two move vectors is calculated. The formulation of this measure θ^k is given below:

$$\theta^k = \frac{\boldsymbol{x}^{k+1} - \boldsymbol{x}^k}{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|} \cdot \frac{\boldsymbol{x}^k - \boldsymbol{x}^{k-1}}{\|\boldsymbol{x}^k - \boldsymbol{x}^{k-1}\|} \tag{16}$$

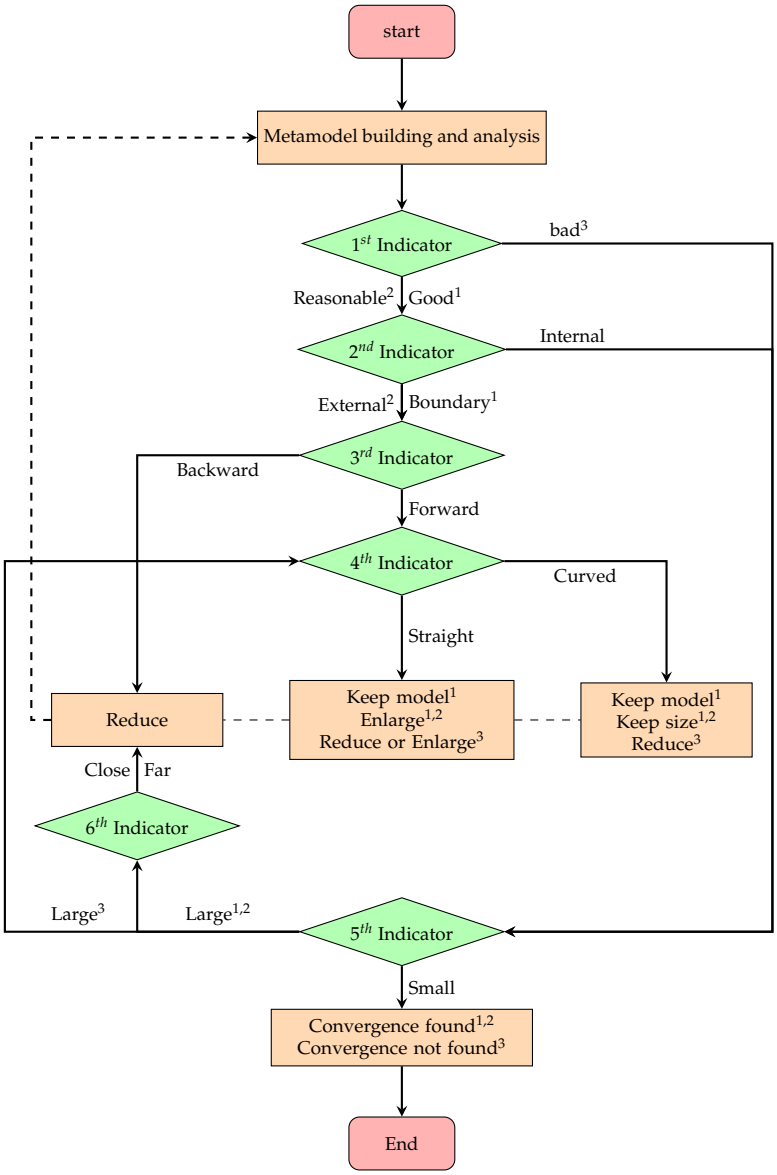

Figure 3. Overview of trust region strategy in MAM

If $\theta^k > 0$ holds, the movement will be denoted as 'Forward', while $\theta^k \leq 0$ is denoted as moving 'Backward'. Moreover, if $\theta^k \leq 0.3$, the convergence history is labelled as 'Curved', otherwise 'Straight'.

The fifth indicator in MAM, as a termination criterion, is the size of the current search subregion. It can be marked as 'Small' or 'Large' according to the quality of the metamodel determined by the first indicator. When the approximations are 'Bad' and $S^k \leq 0.0005$, the present search subregion is considered 'Small'. When the approximations are 'Reasonable' or 'Good', the trust region is denoted as 'Small' if $S^k \leq 0.001$.

The sixth indicator is based on the most active constraint. It is considered to be 'Close' to the boundary between the feasible and infeasible design space if $\boldsymbol{g}_{max}(\boldsymbol{x}^{k+1}) \in [-0.1, 0.1]$, otherwise it is denoted as 'Far'.

Both reduction and enlargement of the trust region is executed using

$$B_i^{k+1} - A_i^{k+1} = \frac{1}{\tau}\left(B_i^k - A_i^k\right) \ (i = 1, ..., d) \tag{17}$$

where τ is the resizing parameter.

When the approximations are 'Bad' and the trust region is 'Small', the current trust region is considered too small for any further reduction to achieve reasonable approximations and the process will be aborted. And when the approximations are 'Bad' and the trust region is 'Large', a reduction of the search region should be applied in order to achieve better approximations. When the approximations are not 'Bad', the trust region is 'Large' and the sub-optimal point is not 'Internal', the 'Backward' convergence history means that the iteration point progresses in a direction opposite to the previous move vector. In this situation, the trust region has to be reduced. And if the iteration point moves 'Forward', and the approximations are 'Good', the same metamodels will be reutilized in the next iteration for the purpose of reducing the computational cost. And if the optimization convergence history is labelled as 'Curved' and the approximations are 'Reasonable', the trust region will be enlarged as the optimization process is moving in the same direction.

A summary of termination criteria as well as the move limit strategy is presented in Table 1 and Figure 3, respectively. Note that in Figure 3, some processes will only be executed when the indicators have the same superscript. For example, the process can only output the final optimum when the approximation is 'Good' (with superscript 1) and the current location (2^{nd} indicator) of the solution is within a small (5^{th} indicator) trust region. If the quality of the metamodel is 'Bad' with the superscript '3' and the 5^{th} indicator has the value 'Large', the 4^{th} indicator will be triggered and a move limit should be then determined.

4.2.3. Space reduction scheme in ASIM framework

As the optimization proceeds, the particles will narrow down their horizon to improve the local search ability. In other words, for each particle involved, the size of the individual trust region will reduce from 1.0 by a factor of 2 in each iteration, i.e. $(\frac{1}{2})^t$ times the size of the initial design space. Although the particles still fly through the whole design space, each individual seems to behave much cleverer and finds the local optimal position more precisely because the metamodel becomes more accurate.

5. Benchmark problem

In this section, the parameters used in MAM and proposed ASIM framework have been given in Table 2 for solving complex optimization problems: one benchmark mathematical example and two real-world complex design problems. The MAM parameters (the maximum number of iteration, the number of required sampling points, the size of the initial trust region and the minimum size of the trust region) are well configured for solving general optimization tasks as proposed in our previous work [28]. And the PSO parameters (the initial weight and the acceleration coefficients) are chosen as the values proposed in [37], which ensure the convergent behavior of the search process.

Table 2: Default parameters for MAM and ASIM

Method	MAM parameters				PSO parameters		
	MI[a]	NOP[b]	SIR[c]	SMR[d]	ω[e]	α[f]	β[f]
MAM	30	$n+5$	0.25	0.1		N.A.	
ASIM	30	$n+5$	0.25	0.1	0.7298	1.49618	1.49618

[a] The maximum number of iteration.
[b] The number of required sampling points.
[c] The size of the initial trust region.
[d] The minimum size of the trust region.
[e] The initial weight in PSO.
[f] The acceleration coefficients in PSO.

5.1. Welded beam

Design optimization of a welded beam in Figure 4 is a complex and challenging problem in nature with many variables and constraints. Usually, conventional optimization methods fail to find global optimal solutions. Hence, the welded beam design problem is often used to evaluate the performance of optimization methods. To determine the best set of design variables for minimizing the total fabrication cost of the structure, the minimum cost optimization is performed considering shear stress (τ), bending stress (σ), buckling load (p_c), and end deflection δ constraints. The design variables comprise the thickness of the weld (x_1), the length of the welded joint (x_2), the width of the beam (x_3) and the thickness of the beam (x_4) and the mathematical formulation of this problem can be expressed as follows:

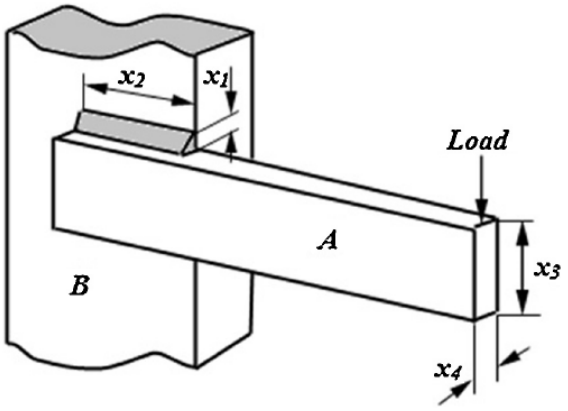

Figure 4. Schematic of the welded beam structure with indication of design variables.

$$
\begin{aligned}
min \quad & f(\boldsymbol{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
s.t. \quad & g_1(\boldsymbol{x}) = \tau(\boldsymbol{x}) - \tau_{max} \leq 0 \\
& g_2(\boldsymbol{x}) = \sigma(\boldsymbol{x}) - \sigma_{max} \leq 0 \\
& g_3(\boldsymbol{x}) = x_1 - x_4 \leq 0 \\
& g_4(\boldsymbol{x}) = [0.10471x_1^2 + 0.04811x_3x_4(14 + x_2)] - 5 \leq 0 \\
& g_5(\boldsymbol{x}) = 0.125 - x_1 \leq 0 \\
& g_6(\boldsymbol{x}) = \delta(\boldsymbol{x}) - \delta_{max} \leq 0 \\
& g_7(\boldsymbol{x}) = p - p_c(\boldsymbol{x}) \leq 0 \\
where \quad & P = 6000\,lb,\ L = 14\,in,\ E = 30 \times 10^6\,psi,\ G = 12 \times 10^6\,psi, \\
& \tau_{max} = 13600\,psi,\ \sigma_{max} = 30000\,psi,\ \delta_{max} = 0.25\,in \\
& \tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right) \\
& R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\
& \tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\
& J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \\
& \sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{Ex_3^3x_4} \\
& p_c(x) = \frac{4.013\sqrt{E\left(\frac{x_3^2x_4^6}{36}\right)}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\
& 0.1 \leq x_1 \leq 2,\ 0.1 \leq x_2 \leq 10,\ 0.1 \leq x_3 \leq 10,\ 0.1 \leq x_4 \leq 2
\end{aligned}
\tag{18}
$$

To solve the aforementioned problem, the GA-based method [39], co-evolutionary PSO method (CPSO) [22], ES-based method [40], charged system search (CSS)[41] and

colliding bodies optimization (CBO) [42] were used to find the optimal solution respectively.

In Table 3, the optimized design variables and cost obtained by MAM and ASIM have been compared with those obtained in literature. The best solutions (1.724852) by MAM and ASIM are more competitive than those obtained by other methods. Although Kaveh [42] claimed 1.724663 was the better cost, the solution actually violated the g_1 constraint and it was an infeasible solution. Based on statistical results in Table 4, it is concluded that the ASIM technique is very robust and efficient because the standard deviation of different runs of simulations is almost 0 ($1.10E-07$) and the number of function analysis (NFEs) is remarkably smaller (565) than that called by other methods except MAM. Both ASIM and MAM demonstrate the efficiency to find the optimal design owing to their accuracy approximations and adaptive trust region strategy at the local level exploitation. Averagely, one hundreds of evaluations are required to determine an optimum. It is noted that the enhancement of the global exploration for the optimal solution by PSO process in ASIM framework could be demonstrated by a standard deviation of zero ($1.10E-07$) for statistical results, which is approximately four orders of magnitude smaller than the value by MAM (0.0031358). Further more, by comparison with the NFEs (200000) obtained by co-evolutionary PSO [22], the accurate surrogates built by ASIM framework indeed assist each particle to find the local refinement position and speed up the converged global optimum. In conclusion, ASIM needs less computational cost for a global optimum with improved accuracy and great robustness.

Table 3: Comparison of present optimized designs with literature for the welded beam.

Methods	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	cost
GA-based [39]	0.205986	3.471328	9.020224	0.20648	1.728226
CPSO [22]	0.202369	3.544214	9.04821	0.205723	1.728024
ES-based [40]	0.199742	3.612060	9.037500	0.206082	1.737300
CSS [41]	0.20582	3.468109	9.038024	0.205723	1.724866
CBO [42]	0.205722	3.47041	9.037276	0.205735	1.724663
MAM	0.2057296	3.4704893	9.0366242	0.2057297	**1.724852**
ASIM	0.2057296	3.4704887	9.0366239	0.2057296	**1.724852**

Table 4: Statistical results from different optimization methods for the welded beam design problem.

Methods	Best	Average	Worst	S.D.	NFEs
GA-based [39]	1.728226	1.792654	1.993408	0.074713	80000
CPSO [22]	1.728024	1.748831	1.782143	0.012926	200000
ES-based [40]	1.737300	1.813290	1.994651	0.070500	25,000
CSS [41]	1.724866	1.739654	1.759479	0.008064	4000
CBO [42]	1.724662	1.725707	1.725059	0.0002437	4000
MAM	**1.724852**	1.725563	1.739605	0.0031358	**122**
ASIM	**1.724852**	**1.724852**	**1.724852**	**1.10E-07**	565

5.2. Design of a tension/compression spring

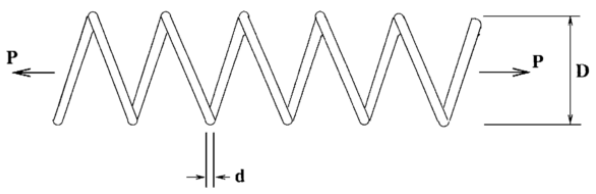

Figure 5. Schematic of the tension/compression spring

This problem first described by Belegundu [43] has arisen from the wide applications of vibration resistant structures in civil engineering. The design objective is

to minimize the weight of a tension/compression spring subject to constraints on the minimum deflection g_1, shear stress g_2, surge frequency g_3 and to limits on the outside diameter g_4. As shown in Figure 5, the design variables include the wire diameter d, the mean coil diameter D, and the number of active coils N. The mathematical description of this problem can be expressed as follows:

$$\begin{aligned} &min\ f(N,D,d) = (N+2)\times Dd^2 \\ &s.t.\ g_1(\boldsymbol{x}) = 1-\frac{D^3N}{71785d^4} \le 0 \\ &\quad g_2(\boldsymbol{x}) = \frac{4D^2-Dd}{12566(Dd^3-d^4)}+\frac{1}{5108d^2}-1 \le 0 \\ &\quad g_3(\boldsymbol{x}) = 1-\frac{140.45d}{D^2N} \le 0 \\ &\quad g_4(\boldsymbol{x}) = \frac{D+d}{1.5}-1 \le 0 \\ &where\ 0.05 \le d \le 1,\ 0.25 \le D \le 1.3,\ 2 \le N \le 15. \end{aligned} \tag{19}$$

The statistical results by MAM are in Table 5. From the first row to the sixth row, every row is the optimal results of 40 independent runs of MAM and the last line concludes the average results of the 6 parallel experiments, i.e., each experiment comprises 40 independent runs of MAM with randomly generated starting points. The best optimal design represented by $[d, D, N]$ is [0.051656122, 0.355902943, 11.33791803] with the objective value of 0.012666692. Moreover, the fourth column 'Best' in Table 5 indicates that MAM can not achieve a converged robust solution and falls into the local optima when faced with multimodal function optimization. The optimal result ranges from 0.01266 (the best design in the fourth row) to 0.070 (the worst design in the third row). As a general deficiency of the trajectory-based algorithm, MAM could not find the known optimum 0.0126652 by balancing the efforts between exploration and exploitation.

A more intuitive perspective for understanding the global search mechanism by ASIM framework has been represented in Table 6, which includes the optimal results obtained by 8 independent experiments, each of which is initialized with 5 particles. In Figure 6, results show the objectives of initial designs and global optima for the tested 40 particles. Even the initial designs are remarkably different at the start of the optimization process due to the random nature of statistical tests, the developed ASIM has the capability to eventually find the converged global optimum. It is concluded that ASIM algorithm can achieve a robust solution for random starting points and it will not be trapped into local optima due to its multi-level search and model management strategies. Therefore, these 8 independent experiments could almost obtain the same global optimum. The best optimal design found by ASIM framework is $[0.051724501, 0.357570887, 11.23912608]$ with the objective value 0.012665259, which has a good agreement with the known optimum. Also, the global solutions from 8 independent experiments have been proved feasible by function evaluations.

Table 5: Statistical results for the tension/compression spring problem by MAM

Number	Worst	Mean	Best	S.D.	NFEs
1	0.032839737	0.015057587	0.0126692	0.004246608	8041
2	0.046478999	0.01537479	0.012677425	0.005275199	8536
3	0.070551755	0.015521846	0.012680762	0.009064574	7483
4	0.053871312	0.016530777	0.012666692	0.00857695	7483
5	0.030829567	0.014687079	0.012733211	0.003455907	7536
6	0.017557055	0.014067046	0.012667273	0.001247161	8149
Average	0.012733211	0.012682427	0.012666692	2.55305E-05	7871

Other algorithms recently used to optimize this problem include: co-evolutionary particle swarm optimization (CPSO) [22], differential evolution with dynamic stochastic

Table 6: Statistical results for the tension/compression spring problem by ASIM

Number	Worst	Mean	Best	S.D.	NFEs
1	0.012707419	0.01268076	0.012669372	1.53792E-05	4891
2	0.015076822	0.013158868	0.012665512	0.001072278	5719
3	0.012734131	0.012681909	0.012665469	2.94215E-05	5161
4	0.013151181	0.012797596	0.012666127	0.000201587	5233
5	0.012674725	0.012671127	0.012665294	3.80784E-06	4882
6	0.012962267	0.012734387	0.012665259	0.000128041	5337
7	0.012787169	0.012679022	0.012669651	1.17008E-05	4702
8	0.012780362	0.01269988	0.012665634	4.68624E-05	5170
Average	0.012669651	0.01266654	0.012665259	1.85492E-06	5141

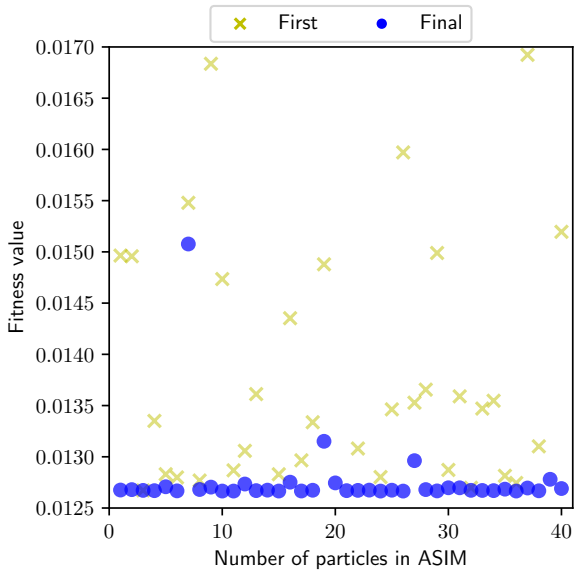

Figure 6. First and final fitness value in ASIM for solving the tension/compression spring

selection (DEDS) [44], hybrid evolutionary algorithm with adaptive constraint-handling techniques (HEAA) [45], league championship algorithm (LCA) [46], water cycle algorithm (WCA) [24] and hybrid cuckoo search (HCS) [23]. The comparison of optimal solutions by aforementioned methods has been given in Table 7 and the statistical results by ASIM, MAM and other algorithms have been shown in Table 8.

Table 7: Comparison of the best solutions obtained by various studies for the tension/compression spring design optimization problem

Name	CPSO [22]	DEDS [44]	HEAA [45]	LCA [46]	WCA [24]	HCS [23]	MAM	ASIM
x_1	0.051728	0.051689	0.051689	0.051689	0.051680	0.051689	0.051656	0.051724
x_2	0.357644	0.356717	0.356729	0.356718	0.356522	0.356718	0.355902	0.357570
x_3	11.244543	11.288965	11.288293	11.28896	11.300410	11.28896	11.33791	11.239126
$g_1(x)$	-8.25E-04	1.45E-09	3.96E-10	N.A./2.00e-15	-1.65E-13	-6.41E-06	$-1.64e-05$	$-1.13e-07$
$g_2(x)$	-2.52E-05	-1.19E-09	-3.59E-10	N.A./-2.22e-15	-7.9E-14	-3.90E-06	$-5.16e-05$	$-1.05e-07$
$g_3(x)$	-4.051306	-4.053785	-4.053808	N.A./-4.053786	-4.053399	-4.053775	−4.051810	−4.055466
$g_4(x)$	-0.727085	-0.727728	-0.727720	N.A./-0.727728	-0.727864	-0.727729	−0.728293	−0.727136
$f(x)$	0.012674	0.012665	0.012665	**0.01266523**	0.012665	**0.0126652**	0.0126667	**0.0126652**

In Table 7, the ASIM framework has the ability to find the optimal solution (0.0126652), which is the best available design, as other algorithms achieved. Although LCA [46] found a slighter better solution (0.01266523), the corresponding constraint $g_1(x)$ was violated. Therefore, it was not a feasible solution. The same conclusion can be drawn for the results by in DEDS [44] and HEAA [45]. Together with the statistical results shown in Table 8, it can be observed that the ASIM method is superior to other methods for the global solution in terms of the number of function evaluations and the accuracy

Table 8: Comparison of statistical results given by different algorithms for the tension/compression spring design optimization problem

Methods	Worst	Mean	Best	S.D.	NFEs
CPSO [22]	0.012924	0.012730	0.012674	5.20E-04	240,000
DEDS [44]	0.012738	0.012669	**0.012665**	1.3E-05	24,000
HEAA [45]	**0.012665**	**0.012665**	**0.012665**	**1.4E-09**	24,000
LCA [46]	0.01266667	0.01266541	**0.01266523**	3.88E-07	15,000
WCA [24]	0.012952	0.012746	**0.012665**	8.06E-05	11,750
HCS [23]	0.0126764	0.0126683	**0.0126652**	5.37E-07	150000
MAM	0.012733211	0.012682427	0.012666692	2.55305E-05	7871
ASIM	0.012669651	0.01266654	**0.012665259**	1.85492E-06	**5141**

throughout the optimization process. Obviously, the referenced methods used more than 10000 calls to find the global optimum while ASIM finds the optimum with about half of those calls. Meanwhile, the ASIM could reduce the number of simulations by over 28% as compared with MAM.

As a general remark on comparisons above, ASIM shows a very competitive performance over eight stat-of-the-art optimization methods to find the global optimal solution in terms of the efficiency, the quality and the robustness.

5.3. Mathematical problem G10

This problem was first described in [47] and then was considered one of the benchmark problems in 2006 IEEE Congress on Evolutionary Computation [48]. In this optimization example, there are eight variables and six inequality constraints (three linear and three non-linear). The mathematical formulations are shown below.

$$\begin{aligned} &min\ f(\boldsymbol{x}) = x_1 + x_2 + x_3 \\ &s.t.\ g_1(\boldsymbol{x}) = -1 + 0.0025(x_4 + x_6) \le 0 \\ &\quad g_2(\boldsymbol{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0 \\ &\quad g_3(\boldsymbol{x}) = -1 + 0.01(x_8 - x_5) \le 0 \\ &\quad g_4(\boldsymbol{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0 \\ &\quad g_5(\boldsymbol{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0 \\ &\quad g_6(\boldsymbol{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0 \\ &where\ 100 \le x_1 \le 10000, 1000 \le x_i \le 10000 (i = 2, 3), \\ &\quad 10 \le x_i \le 1000 (i = 4, ..., 8) \end{aligned} \tag{20}$$

Table 9: Optimal solutions of G10 found by ASIM and MAM

Description	Solution $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]$	Objective value
Known optimum [48]	[579.3066850, 1359.9706780, 5109.970657, 182.0176996, 295.6011737, 217.98230036, 286.4165259, 395.6011737]	7049.2480
ASIM	[579.0697378, 1360.029849, 5110.148583, 181.9979046, 295.5940579, 218.0020914, 286.4038427, 395.5940579]	7049.2481
MAM	[579.2439615, 1360.814966, 5109.189094, 182.0124631, 295.6324343, 217.9875329,286.380036, 395.6324333]	7049.2499

The optimal solutions found by ASIM and MAM are given in Table 9 as well as the known optimum. In Table 10, nine independent experiments have been performed and each experiment includes 40 parallel runs of MAM. Although each run by MAM is initialized with a random starting point, there is no guarantee that the converged global optimum can be achieved. As there has a very small feasible region (0.0010%) in this challenging example, limited runs by MAM could not find a feasible solution and normally a bad design with a very large value of the fitness function (up to 100000)

is obtained. However, a feasible solution could be achieved within 20000 function evaluations. Applying the developed ASIM, the capability of the adaptive multi-level search for the global optimum has been significantly improved and statistical results have been shown in Table 11. Using the same parameter settings in the previous example, the worst solution found by particles is about 7361, which is only 4.42% higher than the global optimum 7049.248. In the mean time, all nine independent experiments of ASIM could find a decent global optimum, which is slightly 10^{-5} higher than the global optimum even in the worst case (Number 5 in Table 11). In Figure 7, it shows how 10 independent runs initialized with total 50 particles converge to the global optimum by ASIM. It is noted that the initial design varies dramatically for each particle, and finally all particles succeed in finding the global optimum. It is concluded that the PSO process applied in ASIM remarkably boosts the exploration capability. Owing to the advantages such as the guidance of personal memory for best position and social cognition, in addition to the stochastic search behavior, ASIM is a robust and efficient algorithm for solving such challenging problem.

Table 10: Statistical results for G10 by MAM

Number	Worst	Mean	Best	S.D.	NFEs
1	142392.7156	17882.82055	7069.390888	30251.49815	18436
2	68065.07371	11619.11583	7049.296446	11320.46339	19388
3	43458.18348	10426.99053	7049.249948	6247.162418	19584
4	76953.38912	12669.27407	7052.442664	12331.32358	18478
5	53761.55465	11938.01326	7060.468503	8513.607171	19122
6	38601.51929	11216.42827	7049.304236	5669.873504	17274
7	133020.3445	12395.33809	7062.698763	19714.98684	19640
8	50195.68872	12527.61721	7061.831868	10079.5634	19668
9	86553.78422	12382.57366	7053.509331	13257.22511	20270
Average	7069.390888	7056.46585	7049.249948	7.334903947	19095

Table 11: Statistical results for G10 by ASIM

Number	Worst	Mean	Best	S.D.	NFEs
1	7071.746167	7054.064714	7049.24996	9.89925027	19374
2	7058.554639	7051.206052	7049.248851	4.112416516	20452
3	7151.048877	7070.606275	7049.248909	45.01472868	19612
4	7361.05089	7112.256788	7049.275307	139.0818573	18450
5	7063.107951	7053.22295	7049.318392	5.982378418	19094
6	7049.802007	7049.418117	7049.248849	0.23813293	19318
7	7361.648467	7111.850254	7049.248177	139.6416717	20102
8	7206.603344	7081.369993	7049.26296	70.01358744	19780
9	7052.697369	7049.999679	7049.251224	1.512797685	19962
10	7105.192042	7060.643287	7049.261412	24.90483007	19794
Average	7049.318392	7049.262676	7049.248177	0.024517331	19522

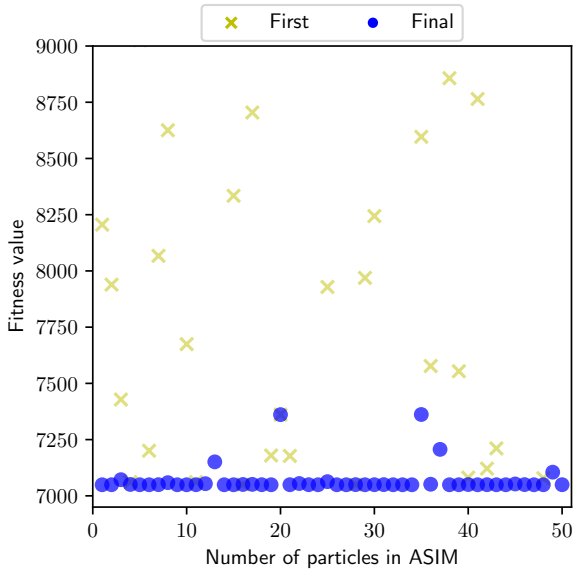

Figure 7. First and final output fitness value for G10 in hybrid optimization framework

Recently, other algorithms including evolutionary optimization by approximate ranking and surrogate models(EOAS) [49], constraint optimization via particle swarm optimization (COPSO) [50], league championship algorithm(LCA) [46], hybrid cuckoo search (HCS) [23], surrogate-assisted differential evolution (SADE) [51] have also solved this optimization problem. A comparison of results by ASIM, MAM and other algorithms has been given in Table 12. Although all methods listed are very competitive and has the ability to find global or near global optimum, ASIM demonstrates the superiority over others in terms of computational efficiency. Evolutionary algorithms usually need over 150000 simulations to find the global optimum while ASIM could reduce the number of function evaluations to 19522 by more than 80%. Further more, the optimum (7049.2481) achieved by ASIM is in a good agreement with the global optimum (7049.2480). Although HCS [23] proposed a best optimum '7049.237', the fourth constraint is slightly violated and therefore that is not a feasible design. Summarily, ASIM outperforms other methods in seeking the global optimal solutions of complex black-box optimization problems in terms of efficiency and accuracy.

Table 12: Statistical features of the results obtained by various algorithms on G10

Methods	Worst	Mean	Best	S.D.	NFEs
EOAS [49]	7258.540	7082.227	7049.404	4.20E+1	304066
COPSO [50]	7049.668593	7049.278821	7049.248871	N.A.	240000
LCA [46]	**7049.2482816**	**7049.2480542**	**7049.2480206**	**5.80E-5**	225000
HCS [23]	7250.957	7049.668	7049.237	8.65E+01	150000
SADE [51]	N.A.	7278.785	7049.249	N.A.	500000
MAM	7069.390888	7056.46585	7049.249948	7.334903947	**19095**
ASIM	7049.318392	7049.262676	7049.248177	0.024517331	19522

6. Conclusion

In this paper, an adaptively integrated swarm intelligence-metamodelling (ASIM) technique, which enables adaptive multi-level adaptive search for the global optimal solution, has been proposed for solving expensive and complex black-box constrained optimization problems. In the first step, the adaptive swarm-intelligence algorithm carries out the global exploration for the near-optimal solution. In the second step, the

metamodel based optimization algorithm, multipoint approximation method (MAM), is performed for the local exploitation. Essentially, each particle's current position in ASIM will gain local refinement by optimization of metamodel building around their neighborhood and tends to move towards to the global best position according to swarm intelligence. Eventually, the swarm as a whole, like a flock of birds collectively foraging for food while each bird is brilliant to find the most tasty food with limited horizon directly, is possibly to move close to a global optimum position. One mathematical problem and two engineering optimization problems are studied in details using ASIM framework. By comparisons of the results obtained from ASIM, MAM and other state-of-art algorithms, it is demonstrated that ASIM has the capability to tackle expensive constrained black-box optimization problems with remarkably less computational effort, higher accuracy and stronger robustness. The adaptive multi-level search ability of ASIM indeed makes up the local search deficiency and the sensitivity to the starting point observed in MAM. Consequently, the ASIM technique achieves a good balance between exploration and exploitation. Moreover, ASIM provides a valuable insight into the development of nature-inspired metaheuristic algorithms for solving nonlinear optimization problems with less computational cost throughout the simulation-based optimization process.

Author Contributions: Dr Guirong Dong contributes to drafting the paper and examples validation; Dr Chengyang Liu contributes to algorithms development and editing; Dr Dianzi Liu contributions to designing and planning the study; approving the final version and agreeing to be accountable for the accuracy and integrity; Dr Xiaoan Mao contributions to editing, analysing and commenting on the first version of manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hickernell, F.J.; Yuan, Y.X. A Simple Multistart Algorithm for Global Optimization. *OR Transactions* **1997**, *1*, 1–12.
2. Rios, L.M.; Sahinidis, N.V. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization* **2013**, *56*, 1247–1293. doi:10.1007/s10898-012-9951-y.
3. Glover, F. Future paths for integer programming and links to artficial intelligence. *Computers and Operations Research* **1986**, *13*, 533–549.
4. Kirkpatrick, S.; Gelatt Jr., C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680.
5. Holland, J.H. *Adaptation in Natural and Artificial Systems: An introductory Analysis with Applications to Biology, Control and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1975; p. 183. doi:10.1137/1018105.
6. Borowska, B. Genetic Learning Particle Swarm Optimization with Interlaced Ring Topology. Computational Science – ICCS 2020; Krzhizhanovskaya, V.V.; Závodszky, G.; Lees, M.H.; Dongarra, J.J.; Sloot, P.M.A.; Brissos, S.; Teixeira, J., Eds.; Springer International Publishing: Cham, 2020; pp. 136–184.
7. Wang, S.; Liu, G.; Gao, M.; Cao, S.; Guo, A.; Wang, J. *Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators*; Vol. 540, Elsevier Inc., 2020; pp. 175–201. doi:10.1016/j.ins.2020.06.027.
8. Ahmad, R.; Choubey, N.S., Review on Image Enhancement Techniques Using Biologically Inspired Artificial Bee Colony Algorithms and Its Variants. In *Biologically Rationalized Computing Techniques For Image Processing Applications*; Hemanth, J.; Emilia, B.V., Eds.; Springer International Publishing: Cham, 2018; pp. 249–271. doi:10.1007/978-3-319-61316-1_11.
9. Acevedo, H.G.S.; Escobar, C.M.; Andres Gonzalez-Estrada, O. Damage detection in a unidimensional truss using the firefly optimization algorithm and finite elements **2018**.
10. Mortazavi, A.; Togan, V. Sizing and layout design of truss structures under dynamic and static constraints with an integrated particle swarm optimization algorithm. *Applied Soft Computing* **2016**, *51*, 239–252. doi:10.1016/j.asoc.2016.11.032.
11. soon Ong, Y.; Nair, P.B.; Keane, A.J.; Wong, K.W. Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems. In *Knowledge Incorporation in Evolutionary Computation*; Yaochu, J., Ed.; Berlin, Germany: Springer-Verlag, 2005; pp. 307–331. doi:10.1007/978-3-540-44511-1_15.
12. Kleijnen, J.P. Kriging metamodeling in simulation: A review. *European Journal of Operational Research* **2009**, *192*, 707–716. doi:10.1016/J.EJOR.2007.10.013.
13. Myers, R.H.; Montgomery, D.T.; Vining, G.G.; Borror, C.M.; Kowalski, S.M. Response Surface Methodology: A Retrospective and Literature Survey. *Journal of Quality Technology* **2004**, *36*, 53–78. doi:10.1080/00224065.2004.11980252.
14. Dyn, N.; Levin, D.; Rippa, S. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM Journal on Scientific and Statistical Computing* **1986**, *7*, 639–659. doi:10.1137/0907043.

15. Gerrard, C.E.; McCall, J.; Coghill, G.M.; Macleod, C. Exploring aspects of cell intelligence with artificial reaction networks. *Soft Computing* **2014**, *18*, 1899–1912. doi:10.1007/s00500-013-1174-8.
16. Dong, H.; Song, B.; Dong, Z.; Wang, P. SCGOSR : Surrogate-based constrained global optimization using space reduction. *Applied Soft Computing Journal* **2018**, *65*, 462–477. doi:10.1016/j.asoc.2018.01.041.
17. Zhou, Z.; Ong, Y.S.; Nair, P.B.; Keane, A.J.; Lum, K.Y. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* **2007**, *37*, 66–76. doi:10.1109/TSMCC.2005.855506.
18. Regis, R.G. Evolutionary Programming for High-Dimensional Constrained Expensive Black-Box Optimization Using Radial Basis Functions. *IEEE Transactions on Evolutionary Computation* **2014**, *18*, 326–347.
19. Bouhlel, M.A.; Bartoli, N.; Regis, R.G.; Morlier, J.; Regis, R.G.; Otsmane, A. Efficient global optimization for high-dimensional constrained problems by using the Kriging models combined with the partial least squares method. *Engineering Optimization* **2018**, pp. 1029–0273. doi:10.1080/0305215X.2017.1419344.
20. Tang, Y.; Chen, J.; Wei, J. A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions. *Engineering Optimization* **2013**, *45*, 557–576. doi:10.1080/0305215X.2012.690759.
21. Regis, R.G. Particle swarm with radial basis function surrogates for expensive black-box optimization. *Journal of Computational Science* **2014**, *5*, 12–23. doi:10.1016/j.jocs.2013.07.004.
22. He, Q.; Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence* **2007**, *20*, 89–99. doi:10.1016/j.engappai.2006.03.003.
23. Long, W.; Liang, X.; Huang, Y.; Chen, Y. An effective hybrid cuckoo search algorithm for constrained global optimization. *Neural Computing and Applications* **2014**, *25*, 911–926. doi:10.1007/s00521-014-1577-1.
24. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures* **2012**, *110-111*, 151–166. doi:10.1016/j.compstruc.2012.07.010.
25. Toropov, V.V.; Filatov, A.A.; Polynkin, A.A. Multiparameter structural optimization using FEM and multipoint explicit approximations. *Structural Optimization* **1993**, *6*, 7–14. doi:10.1007/BF01743169.
26. Keulen, F.V.; Toropov, V.V. New Developments in Structural Optimization Using Adaptive Mesh Refinement and Multipoint Approximations. *Engineering Optimization* **1997**, *29*, 217–234. doi:10.1080/03052159708940994.
27. Polynkin, A.; Toropov, V.V. Mid-range metamodel assembly building based on linear regression for large scale optimization problems. *Structural and Multidisciplinary Optimization* **2012**, *45*, 515–527. doi:10.1007/s00158-011-0692-1.
28. Liu, D.; Toropov, V.V. Implementation of Discrete Capability into the Enhanced Multipoint Approximation Method for Solving Mixed Integer-Continuous Optimization Problems. *International Journal of Computational Methods in Engineering Science and Mechanics* **2016**, *17*. doi:10.1080/15502287.2016.1139013.
29. Caloni, S.; Shahpar, S.; Toropov, V.V. Multi-Disciplinary Design Optimisation of the Cooled Squealer Tip for High Pressure Turbines. *Aerospace* **2018**, *5*. doi:10.3390/aerospace5040116.
30. Toropov, V.V. Simulation approach to structural optimization. *Structural Optimization* **1989**, *1*, 37–46. doi:10.1007/BF01743808.
31. Kraft, D. A Software Package for Sequential Quadratic Programming. Technical report, Institut fuer Dynamik der Flugsysteme, Oberpfaffenhofen, 1988.
32. Kennedy, J.; Eberhart, R. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks. IEEE, 1995, Vol. 4, pp. 1942–1948. doi:10.1109/ICNN.1995.488968.
33. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intelligence* **2007**, *1*, 33–57. doi:10.1007/s11721-007-0002-0.
34. Zhi-Hui Zhan.; Jun Zhang.; Yun Li.; Chung, H.H. Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2009**, *39*, 1362–1381. doi:10.1109/TSMCB.2009.2015956.
35. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation* **2016**, *274*, 292–305. doi:10.1016/j.amc.2015.11.001.
36. Guo, D.; Jin, Y.; Ding, J.; Chai, T. Heterogeneous Ensemble-Based Infill Criterion for Evolutionary Multiobjective Optimization of Expensive Problems. *IEEE Transactions on Cybernetics* **2019**, *49*, 1012–1025. doi:10.1109/TCYB.2018.2794503.
37. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* **2002**, *6*, 58–73. doi:10.1109/4235.985692.
38. Toropov, V.V.; van Keulen, F.; Markine, V.; Alvarez, L. Multipoint approximations based on response surface fitting: a summary of recent developments. Proceedings of the 1st ASMO UK/ISSMO conference on engineering design optimization, Ilkley, West Yorkshire, UK, 1999, pp. 371–381.
39. Coello Coello, C.A.; Montes, E.M.; Mezura Montes, E. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics* **2002**, *16*, 193–203. doi:10.1016/S1474-0346(02)00011-3.
40. Mezura-Montes, E.; Coello Coello, C.A. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems* **2008**, *37*, 443–473. doi:10.1080/03081070701303470.
41. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: charged system search. *Acta Mechanica* **2010**, *213*, 267–289. doi:10.1007/s00707-009-0270-4.
42. Kaveh, A.; Mahdavi, V.R. Colliding bodies optimization: A novel meta-heuristic method. *Computers and Structures* **2014**, *139*, 18–27. doi:10.1016/j.compstruc.2014.04.005.

43. Belegundu, A.D. A study of mathematical programming methods for structural optimization. Ph.d., University of Iowa, 1982.
44. Zhang, M.; Luo, W.; Wang, X. Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences* **2008**, *178*, 3043–3074. doi:10.1016/j.ins.2008.02.014.
45. Wang, Y.; Cai, Z.; Zhou, Y.; Fan, Z. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct Multidisc Optim* **2009**, *37*, 395–413. doi:10.1007/s00158-008-0238-3.
46. Kashan, A.H. An efficient algorithm for constrained global optimization and application to mechanical engineering design : League championship algorithm (LCA). *Computer-Aided Design* **2011**, *43*, 1769–1792. doi:10.1016/j.cad.2011.07.003.
47. Hock, W.; Schittkowski, K. *Test Examples for Nonlinear Programming Codes*; Springer-Verlag Berlin Heidelberg 1981, 1981.
48. Liang, J.; Runarsson, T.; Mezura-Montes, E.; Clerc, M.; Suganthan, P.; A. C. Coello, C.; Deb, K. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. *Nangyang Technological University, Singapore, Tech. Rep* **2006**, *41*.
49. Runarsson, T.P. Constrained Evolutionary Optimization by Approximate Ranking and Surrogate Models. In *Parallel Problem Solving from Nature - PPSN VIII*; Yao, X.; Burke, E.K.; Lozano, J.A.; Smith, J.; Merelo-Guervós, J.J.; Bullinaria, J.A.; Rowe, J.E.; Ti\vno, P.; Kabán, A.; Schwefel, H.P., Eds.; Springer, Berlin, Heidelberg: Berlin, Heidelberg, 2004; pp. 401–410. doi:10.1007/978-3-540-30217-9_41.
50. Aguirre, A.H.; Zavala, A.E.M.; Diharce, E.V.; Rionda, S.B. COPSO : Constrained Optimization via PSO algorithm. Technical report, Center for Research in Mathematics. CIMAT, 2007.
51. Garcia, R.d.P.; de Lima, B.S.L.P.; Lemonge, A.C.d.C. A Surrogate Assisted Differential Evolution to Solve Constrained Optimization Problems. 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI); IEEE: Arequipa, Peru, 2017; pp. 1–6.