



Constraint handling techniques for metaheuristics: a state-of-the-art review and new variants

Nikos D. Lagaros¹ · Makis Kournoutos¹ · Nikos Ath. Kallioras¹ · Alexandros N. Nordas²

Received: 21 April 2022 / Revised: 4 October 2022 / Accepted: 6 October 2022 /

Published online: 17 January 2023

© The Author(s) 2023

Abstract

Metaheuristic optimization algorithms (MOAs) are computational randomized search processes which draw inspiration from physical and biological phenomena, with an application spectrum that extends to numerous fields, ranging from engineering design to economics. MOAs were originally developed for solving unconstrained NP-complete problems, and hence their application to constrained optimization problems (COPs) requires the implementation of specialized techniques that facilitate the treatment of performance and bound constraints. While considerable research efforts have been oriented towards the development and subsequent enhancement of novel constraint handling techniques (CHTs) for MOAs, a systematic review of such techniques has not been conducted hitherto. This work presents a state-of-the-art review on CHTs used with MOAs and proposes eight novel variants based on the feasibility rules and ε -constrained techniques. The distinctive feature of the new variants is that they consider the level and number of constraint violations, besides the objective function value, for selection of individuals within a population. The novel variant performance is evaluated and compared with that of four well-known CHTs from the literature using the metaheuristic pity beetle algorithm, based upon 20 single-objective benchmark COPs.

✉ Nikos D. Lagaros
nlagaros@central.ntua.gr

Makis Kournoutos
s.kournoutos@yahoo.gr

Nikos Ath. Kallioras
nkallioras@yahoo.com

Alexandros N. Nordas
alexandros.nordas@igt.baug.ethz.ch

¹ Institute of Structural Analysis & Antiseismic Research, School of Civil Engineering, National Technical University of Athens, 9, Heroon Polytechniou Str., Zografou Campus, 15780 Athens, Greece

² Department of Civil, Environmental and Geomatic Engineering, ETH Zurich, Stefano-Francini-Platz 5, 8093 Zurich, Switzerland

The computational results highlight the accuracy, effectiveness, and versatility of the novel variants, as well as their performance superiority in comparison with existing techniques, stemming from their distinctive formulation. The complete code can be downloaded from GitHub (<https://github.com/nikoslagaros/MOAs-and-CHTs>).

Keywords Constrained handling techniques · Metaheuristic algorithms · Engineering problems · Optimization computing platform

Abbreviations

COP	Constraint optimization problem
NLP	Non-linear programming
CHT	Constraint handling technique
EA	Evolutionary algorithm
DE	Differential evolution
ORPD	Optimal reactive power dispatch
SA-DECV	Surrogate-assisted differential evolution with combined variants
DCOPs	Dynamic constrained optimization problems
PBA	Pity beetle algorithm
OCF	Optimization computing platform
HP-OCF	High-performance optimization computing platform
APM	Adaptive penalty method
ADP	Automatic dynamic penalization
DSBO	Dynamic surrogate-based optimization
EI	Expected improvement
CHIP	Constraint handling with individual penalty
OPF	Optimal power flow
SR	Stochastic ranking
NPGA	Niched-pareto genetic algorithm
CVI	Constraint violation with interval arithmetic
BSA	Backtracking search algorithm
FOA	Fruit fly optimization algorithm
SAR	Search and Rescue optimization algorithm
WDOCHM-PSO	PSO-based algorithm combined with dynamic objective constraint handling method
DFIG	Doubly fed induction generator
SCE	Shuffled complex evolution
CCE	Competitive complex evolution
EP	Evolutionary programming
FS	Feasible solutions
SPSA	Simultaneous perturbation stochastic approximation
CMA-ES	Covariance matrix adaptation evolution strategy
MOA	Metaheuristic optimization algorithm
RST	Random sampling technique

1 Introduction

Most modern optimization problems are pronouncedly nonlinear and contain variables subject to several function and bound constraints. Such problems are called constrained optimization problems (COPs) or nonlinear programming (NLP) problems and their formulation can be described in the following general form, according to Deb (2000):

$$\begin{aligned}
 & \text{Minimize } f(x), \\
 & \text{Subject :} \\
 & g_j(x) \leq 0, \quad j = 1, 2, \dots, J, \\
 & h_k(x) = 0, \quad k = 1, 2, \dots, K, \\
 & x_i^l \leq x_i \leq x_i^u, \quad i = 1, 2, \dots, n,
 \end{aligned} \tag{1}$$

where $f(x)$ is the objective function of the vector variable x , $g_j(x)$ are inequality and $h_k(x)$ are equality constraints, both referred to also as performance constraints, and x_i^l, x_i^u are respectively the lower and upper limit values of component x_i in x . Equality constraints can be converted into inequality constraints as $g_k(x) = |h_k(x)| - e \leq 0, k = 1, 2, \dots, K$, where e is a small positive quantity (e.g. 1.0E-3), and hence the two groups of performance constraints in expression (1) can be unified into a single group $g_j(x) \leq 0, j = 1, 2, \dots, m$, where $m = J + K$.

Metaheuristic optimization algorithms (MOAs) are computational randomized search processes which draw inspiration from physical and biological phenomena. Over the years, the application field of MOAs has expanded vastly beyond physical and biological sciences and into different fields, ranging from engineering to economics. Nonetheless, most MOAs were originally developed for solving unconstrained NP-complete optimization problems, and hence their application to COPs is rendered a challenging undertaking due to the posed constraint handling requirement. However, most of the problems where MOAs are used concern COPs. The effective overcoming of the constraint handling obstacle has progressively evolved into an independent research field, particularly since a unique constraint handling technique (CHT) which guarantees search algorithm convergence to the global optimum does not exist. The focus of research efforts has been shared between development of novel CHTs and performance enhancement of existing CHTs. With focus on MOAs, several authors have developed, enhanced, and evaluated different CHTs implemented in specific MOAs, for application to COPs across a broad range of fields.

Coello and Montes (2002) conducted a detailed review and performance evaluation of the most popular CHTs implemented in evolutionary algorithms (EAs), including penalty, separation of constraints and objectives, special operators, hybrid and repair-algorithms-based CHTs. The authors further evaluated several penalty-based techniques implemented in a genetic algorithm (GA). A detailed review on several CHTs integrated into nature-inspired algorithms, e.g. EAs, or based on swarm intelligence, was conducted by Mezura-Montes and Coello (2011), including feasibility rules, ϵ -constrained, penalty, stochastic ranking, special operators, multi-objective concepts and ensemble CHTs. Focusing on the solution of optimal reactive power

dispatch (ORPD) problems, Mallipeddi et al. (2012) evaluated penalty and adaptive penalty, superiority of feasible solutions, ε -constraint method, stochastic ranking, and ensemble CHTs, implemented in differential evolution (DE) algorithms. With focus on particle swarm optimization (PSO), the applicability and performance of penalty-based CHTs, separatists, hybrids of PSO algorithm with other optimization techniques, as well as other methods not falling under the aforementioned categories was assessed by Jordehi (2015). Miranda-Varela and Mezura-Montes (2018) evaluated four surrogate-assisted differential evolution with combined variants (SA-DECV) algorithms combined with feasibility rules, ε -constrained method, stochastic ranking, and diversity maintenance CHTs, on the basis of 24 well known test functions, and conducted comparisons with other surrogate-assisted algorithms. Ameca-Alducin et al. (2018) evaluated a DE algorithm combined with penalty function, feasibility rules, ε -constrained method and stochastic ranking CHTs, based upon benchmark dynamic COPs. Lin et al. (2019) assessed the performance of four different dynamic multimodal population-based optimization algorithms with superiority of feasible solution, ε -constraint method, penalty function, dynamic penalty function and stochastic selection and ranking CHTs, in the context of dynamic multimodal COPs. Caraffini et al. (2019) analyzed the behavior of various popular DE schemes combined with different CHTs, such as penalty function and correction techniques (saturation, toroidal), with focus on establishing the mutation and crossover operators that introduce structural bias in DE algorithms and appropriate combinations of CHTs, DE control parameters and population size to moderate the bias.

Notwithstanding the substantial research outlined above, a systematic review of existing CHTs for MOA has not been conducted hitherto. Within this backdrop, this paper aims to provide a detailed and complete state-of-the-art review of the most widely employed and efficient existing CHTs for MOAs. Moreover, the paper proposes eight novel variants of the well-known feasibility rules and ε -constrained CHTs and assesses their performance in comparison with existing CHTs using a swarm intelligence piny beetle algorithm (PBA) (Kallioras et al. 2018), based upon 14 mathematical and 6 engineering COPs. All computations are performed using the MATLAB variant of High-Performance Optimization Computing Platform (HP-OCP), the evolution of the original OCP (Lagaros 2014); OCP is founded on an object-oriented general-purpose code in C#, specifically developed for structural design optimization in civil engineering applications.

The paper proceeds with providing a comprehensive state-of-the-art review of almost 60 CHT studies from the literature in Sect. 2. In Sect. 3, an overview of the penalty methods, feasibility rules, ε -constrained, stochastic ranking, and ensemble CHTs is provided and eight novel variants of the feasibility rules and ε -constrained CHTs are introduced. Subsequently, a brief description of HP-OCP and PBA is provided in Sect. 4. Finally, a comparative performance evaluation of the existing and novel CHTs presented in Sect. 3 is conducted in Sect. 5, based upon 14 mathematical and 6 engineering COP numerical examples, followed by concluding remarks in Sect. 6.

2 Comprehensive review on constraint handling techniques

This section provides a comprehensive review on the most widely employed and efficient existing CHTs for MOAs, classified into six categories: (i) penalty-based CHTs; (ii) CHTs based on separation of objective and constraints; (iii) combination of CHTs; (iv) repair-algorithm-based CHTs; (v) boundary-based CHTs; and (vi) other CHTs. Table 1 provides a summary of all reviewed CHTs, along with the MOA employed in the respective work and the corresponding literature reference number.

2.1 Penalty-based constraint handling techniques

Objective function penalization is the oldest and yet most widely employed CHT. Several penalty-based techniques have been developed over the years, including, among others, the static, dynamic, adaptive and death penalty methods.

During the early 2000s, Miettinen et al. (2003) tested five penalty-based CHTs with genetic algorithms to solve 33 mathematical test problems of different types, concluding that the adaptive penalty method (APM) was the most efficient and the parameter free penalty method the most reliable among them.

Nearly a decade later, Da Silva et al. (2011) proposed a parameter-free APM for use with DUVDE algorithm, a DE that incorporates a mechanism named Dynamic Use of Variants (DUV) that tries and automatically selects the variant with the best performance during the procedure, in the context of structural and mechanical engineering applications. This CHT, by using the feedback received from the current status of candidate solution population to adaptively define the corresponding penalty factor for every constraint combined with DUV, made DE more efficient in solving COPs and achieved very competitive results compared with GA and simple DE.

Montemurro et al. (2013) proposed a parameter-free penalty-based CHT called automatic dynamic penalization (ADP) and applied it to practical engineering design problems, e.g. maximization of the first buckling load of composite laminates. The proposed CHT is combined with a multi-population genetic algorithm named BIANCA, where simultaneous evolution of different species individuals takes place by utilizing special operators for crossover and mutation. ADP exploits the information of infeasible individuals, automatically selects, and updates the penalty factors, thus enabling a wide exploration of the search space.

In Li et al. (2019a) employed a Kriging model in a dynamic surrogate-based optimization (DSBO) of standard test functions and an engineering problem, utilizing two criteria for sample selection: maximization of the expected improvement (EI) function; and minimization of surrogate model prediction, and achieved constraint handling in three different ways: (a) constraining EI function, (b) penalizing the prediction of the surrogate model and (c) penalizing the objective function. DSBO achieved best performance in two tests and a beam optimization problem with the simultaneous employment of methods (a) and (b).

The same year, Kawachi et al. (2019) introduced a DE algorithm named LSHADE equipped with an APM CHT, which is not based solely on the objective function value or the constraint violation in the search region, but also considers their balance, and

Table 1 Overview and classification of constraint handling techniques (CHTs) for metaheuristic optimization algorithms

CHT Class	CHT Variant	Reference Number	Metaheuristic Optimization Algorithm	
Penalty-based	Different penalty methods	12	Genetic Algorithms (GA)	
	Adaptive penalty (APM)	13	Differential Evolution (DUVDE)	
	Automatic Dynamic Penalization (ADP)	14	Genetic Algorithms (BIANCA)	
	Constraining EI function, penalizing surrogate prediction and penalizing objective function	15	Dynamic Surrogate-based optimization (DSBO)	
	Adaptive penalty	16	Differential Evolution (LSHADE)	
	Individual penalty	17	Genetic algorithms (NSGA-II)	
	Self-adaptive penalty	18	Differential Evolution (EJADE)	
	Separation of objective and constraints	Feasibility rules	1	Genetic Algorithms (GA)
		Stochastic ranking	19	Evolution Strategies (ES)
		Multi-objective concept	20	Genetic algorithms (NPGA)
		Adaptive constraint-handling technique (ACT)	21	Hybrid Evolutionary algorithm (HEA)
		ϵ -constrained method	22	Differential Evolution (DE)
		Constraint violation with interval arithmetic (CVI)	23	Particle Swarm Optimization (PSO)
		Feasibility rules	24	Evolutionary Algorithms (Backtracking Search Algorithm)
		ϵ -constrained method with fixed control way of ϵ value		
		ϵ -constrained method with self-adaptive control way of ϵ value		
		VCH (Violation Constraint-Handling) technique	25	Genetic Algorithms (GA)

Table 1 (continued)

CHT Class	CHT Variant	Reference Number	Metaheuristic Optimization Algorithm
Combination of Constraint Handling Techniques	Biased dynamic weights	26	Evolutionary Algorithms (DE based algorithm)
	Improved ϵ -Constrained method (IEpsilon)	27	Differential Evolution (L-SHADE44)
	Extended balanced ranking method (E-BRM)	28	Evolutionary Algorithms
	Fuzzy search bias (FSB)	29	Evolutionary Algorithms (DE, ES, GA, Particle Swarm Optimization (PSO) and Differential Evolution (SADE))
	Feasibility rule with the incorporation of objective function information (PROFI)	30	
	Improved Deb (IDeb) rules	31	Fruit fly optimization algorithm (FOA)
	ϵ -constrained method	32	Differential Evolution (DE)
	ϵ -constrained method	33	Search and Rescue optimization algorithm (SAR)
	ϵ -constrained method	34	Differential Evolution (LSHADE)
	Weighted Dynamic-Objective Constraint-Handling Method (WDOCHM)	35	Particle Swarm Optimization (PSO)
	Stochastic ranking	36	Shuffled Complex Evolution (SCE)
Feasibility rules, self-adaptive penalty, ϵ -constraint method and stochastic ranking	37	Differential Evolution (DE) and Evolutionary Programming (EP)	
Penalty method and feasibility rules	38	Differential Evolution (UDE)	
Feasibility rules and self-adaptive penalty	39	Differential Evolution (DE)	
Landscape-Aware Constraint Handling	40	Differential Evolution (DE)	
Feasibility rules and ϵ -constraint method	41	Differential Evolution (DEVNS)	

Table 1 (continued)

CHT Class	CHT Variant	Reference Number	Metaheuristic Optimization Algorithm
Repair algorithm based	Feasibility rules, self-adaptive penalty, ϵ -constraint method and stochastic ranking	42	Differential Evolution (JADE, SADE)
	Feasibility rules and repair algorithm	43	Particle Swarm Optimization (PSO)
	Gradient-based repair method	44	Genetic Algorithms (GA)
	Different repair methods	45	Evolutionary Algorithms (GA based one)
	Repair operator	46	Genetic Algorithms (GA)
	Modified repair process based on incremental rate	47	Particle Swarm Optimization (PSO) and Differential Evolution (DE)
	Boundary update (BU) method	48	Mathematical Interior-Point (IP), Sequential Quadratic Programming (SQP), Direct Search (DS)) and Metaheuristic Algorithms (Simulated Annealing (SA), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Evolution (DE))
	Evolutionary boundary constraint handling (EBCH) scheme	49	Differential Evolution (DE)
Boundary based	Probabilistic evolutionary boundary constraint handling (PEBCH) method	50	Particle Swarm Optimization (PSO)
	Boundary constraint handling scheme	51	Chaotic Evolutionary Algorithms
	Different boundary constraint handling methods	52	Differential Evolution (DE)
	Adaptive boundary constraint-handling scheme (ABCHS)	53	Particle Swarm Optimization (PSO) and Differential Evolution (DE)

Table 1 (continued)

CHT Class	CHT Variant	Reference Number	Metaheuristic Optimization Algorithm
Other	Projection method and the Logarithmic Transformation method	54	Gradient-based algorithms
	Different boundary constraint handling methods	55	Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
	Augmented Lagrangian method	56	Evolution Strategies (ES)
	SCU-CI strategy	57	Genetic Algorithms (GA)
	PSO-SVM Approach	58	Particle Swarm Optimization (PSO)
	Enhanced PSO with a Multi-Strategy Implementation (PSO-MS)	59	Particle Swarm Optimization (PSO)

applied it to the optimization of benchmark functions provided by the Congress on Evolutionary Computation (CEC) 2017. Using the right balance between constraint violation and objective function values, the proposed method outperformed a conventional APM CHT, as well as other CHTs, in most cases considered.

Also, that year Datta et al. (2019) proposed the constraint handling with individual penalty (CHIP) technique, which is a combination of a common penalty-based CHT with the bi-objective EA NSGA-II. The method considers an individual penalty factor for each constraint, instead of utilizing a global factor for the overall constraint violation; nonetheless, the overall constraint violation is employed as an auxiliary objective function subject to minimization, to adaptively compute the individual penalty factors. The benefits of the proposed CHT are the capability of constraint automatic normalization and the low associated computational demand.

In Li et al. (2020) introduced the EJADE-SP algorithm for application to optimal power flow (OPF) problems, which is an enhanced adaptive DE combined with a self-adaptive penalty (SP) CHT that includes many enhancements, e.g. strategy for dynamic population reduction, crossover rate (CR) sorting mechanism, re-randomizing of CR and scale factor F parameters. The combination of this algorithm with an SP CHT achieved comparable performance to other representative algorithms such as simple DE combined with different CHTs.

2.2 Techniques based on separation of objective and constraints

In Deb (2000) developed a penalty-based CHT called feasibility rules, applicable only to population-based algorithms, e.g. genetic or other EAs, which requires no penalty parameters and is based on comparisons between pairs of individuals according to the so-called feasibility rules used in tournament selection. The major advantage of the proposed method is the elimination of the requirement of appropriate penalty factor selection by the user, which is otherwise a necessity in penalty-based CHTs to guide the search to the optimum solution.

In the same year, Runarsson and Yao (2000) introduced a novel CHT called stochastic ranking (SR) which considers the balance between objective and penalty functions via a stochastic bubble-sort algorithm and used it within an evolution strategy to solve 13 benchmark problems. Like the method proposed by Deb (2000), SR also offers the considerable advantage of eliminating the requirement of penalty factor selection by the user.

In Coello and Mezura-Montes (2002) employed within the context of structural design optimization problems the niched-pareto genetic algorithm (NPGA), which performs tournament selection based on the principle of nondomination frequently employed in multi-objective problems. For the NPGA application to single-objective COPs, the constraints are handled as additional objective functions. Using the multi-objective concept, the proposed method handles constraints efficiently without using penalty functions, while also maintaining the diversity in the population without any niching.

In Wang et al. (2009) introduced a hybrid EA that combines two mutation operators with a simplex crossover operator to generate new offsprings and employed it in

13 well-known benchmark test problems. Within the algorithm an adaptive CHT is implemented, which considers a different CHT depending on the population status based on infeasible, semi-feasible and feasible conditions. The proposed method has numerous benefits, including simplicity of implementation, enhanced robustness, and effectiveness.

In Takahama and Sakai (2010), proposed a DE algorithm with an archive and gradient-based mutation operator combined with an ε -constrained CHT, and successfully applied it to 18 test problems from CEC2010. The proposed CHT conducts ε -level comparisons between individuals with automatic control parameter adjustment, to relax the constraints and keep useful information from infeasible individuals during the search process.

In Mazhoud et al. (2013) examined a CHT called constraint violation with interval arithmetic (CVI) and employed it with a customized PSO algorithm in 24 benchmark mathematical and 3 engineering optimization problems. CVI employs the total constraint violation as an additional objective function subject to minimization, interval arithmetic for total violation normalization, as well as a simple lexicographic method for solving subsequent problems.

In Zhang et al. (2015) implemented in a new EA called backtracking search algorithm (BSA) different CHTs, such as feasibility rules and ε -constrained method with two ways of controlling the ε value and applied it to 13 benchmark functions and 4 engineering problems. The ε -constrained CHT with self-adaptive control of the ε value exhibited superior performance in respect of efficiency and convergence speed compared to the other two CHTs.

In Chehouri et al. (2016) used a GA as a numerical tool to implement a parameter-free CHT for application to NLP problems. The approach is based on the use of the violation factor and entails the population distinction into two families of feasible and infeasible solutions, where the latter contains individuals violating at least one constraint; for the population evolution at each generation, the first family is sorted with respect to the value of the fitness function, while the second according to comparison rules (feasibility-based rules). The distinctive feature of these rules is the additional consideration of the number of violated constraints, beyond the constraints violation value, in the comparison of two infeasible individuals.

In Peng et al. (2018) employed EAs combined with a CHT based on biased dynamic weights and applied it to 24 CEC2006 and 18 CEC2010 benchmark problems. Biased weights enable the selection of individuals with low objective and constraints violation values, following which the weights are dynamically adjusted to focus mostly on those individuals. In this manner, the approach prioritizes the selection of infeasible individuals from the population of parents and children, which are closer to the feasible search space. This CHT has been shown to exhibit a stable, reliable, and competitive performance when integrated in a DE algorithm.

In Fan et al. (2018) integrated an improved ε -constrained CHT called IEpsilon into DE algorithm LSHADE44 and applied it to 28 CEC2017 benchmark problems. The distinctive feature of IEpsilon compared to the classic ε -constrained handling method is the adaptive control of the ε value with respect to the proportion of feasible individuals in the current population, which balances the search between feasible and infeasible space during the evolutionary process.

In Rodrigues et al. (2018) proposed an alternative version of the balanced ranking CHT method, called Extended-BRM, and implemented it in evolutionary algorithms for application to CEC2006 and CEC2010 test functions and engineering problems. The proposed technique employs a self-adaptive mechanism to rank feasible and infeasible individuals in two separate groups, and later merges them during the search process with respect to the proportion of feasible and infeasible individuals in the current population. This CHT requires no parameter adjustment by the user, produces the most feasible solutions, and achieves superior performance compared to other CHTs, such as stochastic ranking, adaptive penalty method, etc.

In Li et al. (2019b) employed a CHT based on fuzzy knowledge named FSB, guided by a membership function, in conjunction with EAs to optimize 24 benchmark test functions. Fuzzy knowledge is based on the principle that candidate solutions with large constraints violation values lead the search to candidates with smaller constraints violation values which, in turn, leads to candidates with better objective function value. The FSB CHT integrated into different EAs (DE, ES, GA, PSO, EP) proved to be more robust and efficient compared to penalty function, stochastic ranking, ϵ -constrained and feasibility rules CHTs.

Recently, Zhao et al. (2020) introduced a feasibility-rule-based CHT named FROFI, integrated into a radial basis function (RBF) surrogate-assisted DE algorithm, and applied to 2 test problems and 2 reservoir models. FROFI integrates information from the objective function through the DE algorithm operators, along with a replacement procedure, into the well-known Deb rules, to achieve a more efficient balance between objective function and constraints.

Based on the premise that among two solutions the one with better fitness function value is preferred, in Liu et al. (2020) integrated an improved Deb rule called IDEb into a Fruit fly optimization algorithm (FOA) (a search strategy based on memory, inspired by the foraging behavior of fruit flies) for application to the optimization of truss structures. The method initially employs the classic Deb rule to identify feasible individuals in the solutions memory size and subsequently evaluates the constraints violation, alas only of the individuals with better objective function values compared to the worst one stored in memory. Using the IDEb CHT a substantial reduction in the computational demand associated with structural analysis procedures can be achieved, since in such problems constraints violation evaluations are more computationally expensive compared to objective function evaluations.

In the same year, Chu et al. (2020) employed an ϵ -constraint CHT in a DE algorithm for topology optimization of polyline-based core sandwich structures (PCSSs). Amongst the volume, angle, and thickness constraints to be fulfilled, the predominant volume constraint is approximated by the RBF surrogate model for low computational cost, at first, and then handled through the EC method to compare different infeasible solutions; in this manner a more effective search space exploration is achieved, while potential errors arising from the prediction model RBF are reduced.

Shabani et al. (2020) proposed a metaheuristic search and rescue (SAR) optimization algorithm, which simulates people behavior during search and rescue operations. A high performance was achieved by SAR when combined with an ϵ -constrained CHT to several benchmark test functions and 7 engineering problems.

In the work by Stanovov et al. (2020), several variations of the ε -constraint CHT are used with a DE algorithm to optimize benchmark functions and economic load dispatch problems. Among the classical ε -constraint CHT, the proposed variants of ε -constraint CHT, and the Epsilon method, the most efficient was the combined fitness– ε vector length with individual penalty levels (EC_{IPL}) variant, which considers both constraints violation and objective function values, as well as each constraint individually.

Rodrigues et al. (2020) proposed a PSO based algorithm combined with a weighted dynamic objective constraint handling method (WDOCHM-PSO) for design optimization of doubly fed induction generators (DFIGs). This method entails splitting the constrained problem into two unconstrained objective functions: the first considers with the distance of infeasible individuals from the feasible region (i.e. violation); and the second considers the optimization problem without constraints and is used only for feasible individuals.

Mao et al. (2019) adopted the SR CHT integrated into an optimization algorithm called shuffled complex evolution (SCE), which combines the competitive complex evolution (CCE) algorithm with the simplex method to examine constrained reservoir scheduling problems. The SR CHT has been shown to be highly efficient in the identification of feasible solutions, even in the early steps of the optimization process, while its integration into SCE results in a highly robust algorithm, capable of detecting feasible regions swift.

2.3 Combination of constraint handling techniques

A popular trend in recent years is the combination of different CHTs during the optimization process. In Mallipeddi and Suganthan (2010) employed an ensemble of CHTs (feasibility rules, self-adaptive penalty, ε -constraint method, and stochastic ranking) combined with the Evolutionary Programming (EP) and DE algorithms in the context of COPs. In the proposed approach each CHT refers to a different population, alas every parent individual is compared with the produced children of all populations. A combination of different CHTs achieved superior performance compared to individual CHTs.

In Trivedi et al. (2017) adopted an ensemble of the penalty method and feasibility rules CHTs, integrated in a Unified Differential Evolution (UDE) algorithm, for the optimization of 28 CEC 2017 test problems. The proposed approach applies penalization to the first half of maximum allowed function evaluations, for efficient exploration of the search space, and feasibility rules to the other half FEs, for efficient exploitation.

In Biswas et al. (2018) examined standard IEEE 30 57- and 118-bus systems for OPF objective functions, including cost, emission, voltage stability etc., using a self-adaptive penalty CHT, a feasibility rules CHT, as well as an ensemble of these, integrated within a DE algorithm. In most cases, a combination of CHTs outperformed individual CHTs.

In the same year, Malan (2018) used a landscape-aware constraint handling approach integrated in a DE algorithm for the optimization of 18 CEC2010 test problems. The distinguishing feature of the landscape-aware approach compared to well-known ensemble of CHTs is the algorithm switch between four CHTs based on

the characteristics of the landscape gathered during the search process, with only one CHT being active at a time.

In the work by Wang et al. (2019) in 2019, a DE algorithm variant called DEVNS, which produces each trial individual using two special mutation strategies and random crossover and mutation values, is combined with feasibility rules and ε -constrained CHTs in the optimization of several CEC2017 test problems. In the proposed approach, feasibility rules are applied for comparison of trial and target individuals, however all individuals with lower constraint violation than the specified ε level are considered feasible. In this manner a relaxation of constraints is achieved, and hence a more efficient search.

Also in 2019, Javed et al. (2019) used an ensemble of feasibility rules, self-adaptive penalty, ε -constrained and stochastic ranking CHTs, integrated into JADE and SADE algorithms, in the optimization of 24 CEC2006 benchmark problems. Both algorithms achieved very competitive feasibility rate, alas inferior success rate, compared to other considered algorithms employing a single CHT, based upon the CEC2006 evaluation criteria.

With focus on numerical optimization problems, Kaucic et al. (2020) introduced a PSO-based algorithm called IC-PSO, combined with a hybrid CHT which combines Deb rules with a correction mechanism. The proposed CHT utilizes a suitable operator for repairing infeasible individuals to become feasible and subsequently applies feasibility rules, thus accelerating the PSO algorithm convergence to feasible regions.

2.4 Repair algorithm-based constraint handling techniques

In the solution of COPs, a specific procedure (typically heuristic) called repair algorithm is implemented with the aim to repair an infeasible individual, i.e. convert it to a feasible one. The repaired individual can then be used solely for fitness function evaluation purposes or for replacing the initial individual into the next generation during the evolutionary process. The use of the repaired individual depends principally on the nature of the problem and the developer decision and varies between either replacing the original individual with the repaired one, or only a small percentage, or every infeasible individual in the population during the evolutionary process.

In Chootinan and Chen (2006) integrated into a simple GA a repair mechanism based on the constraint gradient information, to solve 11 optimization test problems. Constraint gradient information is obtained either via approximate finite differences or direct derivatives of the constraints, with the aim of guiding the infeasible individuals into the feasible search space. The considered CHT requires only a repair probability parameter adjustment by the user and has been shown to achieve comparable performance to other considered CHTs, such as stochastic ranking.

In the work by Salcedo-Sanz (2009), a comprehensive survey of the main repair heuristics used with EAs is conducted, to examine their performance in COPs.

Recently, Zein et al. (2017) employed a repair operator integrated into a GA to perform in the context of a composite structure preliminary design. The approach considers an inner and an outer loop, where the inner solves the problem using the GA and the surrogate models, while the outer reconstructs the surrogate model according

to the optimal surrogate solution. The proposed method has been shown to achieve a good balance between accuracy and computational demand.

In Li et al. (2018) integrated in PSO and DE algorithms a modified repair procedure, to handle the equality constraints on Economic Dispatch problems using six different test generators systems. The repair technique evaluates the objective function derivative and uses it to adapt incrementally the output of the unit by sharing the unbalanced amount of the system constraint violation. The employed CHT has been shown to be very efficient, mainly for large-scale power systems.

Gandomi and Deb (2020) in 2020 implemented a CHT called boundary update (BU) method in EAs and mathematical algorithms and tested it in several engineering and mathematical optimization problems. BU repairs the limits of selected variables to avoid constraints violation by restricting and changing them during the optimization process, generating solutions within the search space “updated” limits. Despite the BU method requiring constraint variables pre-categorizing, it achieves a high performance in the solution of COPs with complex constraints.

2.5 Boundary-based constraint handling techniques

The constraints in expression (1) can be distinguished into function constraints and bound constraints. Function constraints encompass inequality or equality functional forms of the decision variables, while bound constraints directly enforce upper and lower values (bounds) to the decision variables. While most of the research has focused on the development of CHTs for function constraint handling, a few significant research efforts have concentrated on the development of Boundary Constraint Handling Methods (BCHMs). BCHMs modify and correct the position of infeasible variable solution vectors, to reset them in the search space and convert them into feasible.

In the work by Gandomi and Yang in (2012) a comparison of an evolutionary boundary constraint handling (EBCH) scheme integrated into DE with other boundary-based methods is conducted, based upon several test problems. In the proposed method, individual components that violate a bound are replaced by a random component, which lies in between the violated bound and the corresponding component of the current best solution. The proposed scheme achieves very good performance, in most cases surpassing other classical BCHMs, such as absorbing, toroidal, reflecting, and random.

In Gandomi and Kashani (2018) proposed a probabilistic evolutionary boundary constraint handling (PEBCH) method combined with PSO for optimizing 7 benchmark test functions. PEBCH is a probabilistic version of EBCH (Gandomi and Yang 2012), which additionally considers the boundaries violation quantity. Specifically, a probability distribution function is employed, the density of which varies depending on the distance of the violated solution and boundary, taking higher values close to the violated boundary.

The results of utilizing various maps based on chaotic behavior within evolutionary algorithms (EAs) are investigated by Mozaffari et al. (2019). To handle the boundary

constraints, a controlling formula which transfers a violated solution into the acceptable solution limits is used to achieve the correction of a violated solution position; in this manner global search is enhanced and early convergence is prevented.

In Biedrzycki et al. (2019) conducted a performance assessment of various combinations between seven DE algorithms and seventeen BCHMs, based upon CEC2017 benchmark functions, concluding that the efficiency of DE algorithms varies substantially depending on the employed BCHM.

In the same year, Juárez-Castillo et al. (2019) implemented an adaptive BCH scheme in PSO and DE algorithms for the optimization of sixty single-objective COPs. The adaptive approach employs a specific BCHM if the population consists entirely of infeasible solutions, otherwise a BCHM from a predefined set is randomly selected, based on a dynamically-updated probability associated with it. This adaptive scheme achieved better performance compared to single BCHMs in COPs with moderate or large number of variables or constraints, as well as in COPs where the identification of feasible individuals is rendered cumbersome.

In Arouri and Sayyafzadeh (2020) developed a gradient-based algorithm that combines features of simultaneous perturbation stochastic approximation (SPSA) and an adaptive moment estimation framework, for application to production optimization problems. The algorithm was tested using the projection and logarithmic transformation BCHMs, with the former achieving superior performance compared to the latter in both cases examined.

In the work by Biedrzycki et al. in the same year (2020), the performance of the covariance matrix adaptation evolution strategy (CMA-ES) algorithm is assessed in the optimization of unimodal and multimodal CEC2017 and Black-Box Optimization Benchmarking (BBOB) problems, considering 22 BCHMs categorized as repair, feasibility preserving, specialized and penalty functions. Amongst all considered BCHMs, Darwinian reflection and resampling achieved the highest performance.

2.6 Other constraint handling techniques

In Atamna et al. (2020) implemented an augmented Lagrangian approach as CHT in Evolution Strategies for optimizing sphere and ellipsoid functions with linear constraints, where the initial constrained objective function is converted into an unconstrained augmented Lagrangian function, while its parameters are adopted during optimization.

In the same year, Qian et al. (2020) employed a GA combined with a surrogate model for solving 9 numerical and 2 engineering optimization problems, using the Kriging surrogate prediction model instead of the actual constraints, which is continuously updated during the optimization procedure to avoid infeasible optimization solutions. This method leads to optimal feasible solutions and is associated with a low computational demand compared to other existing techniques.

Rosso et al. (2021) presented a new nonpenalty-based constraint handling approach for PSO type of algorithms adopting support vector machine (SVM), that is a supervised classification machine learning approach. Due to its generality, constraint

handling with SVM appears more adaptive both to nonlinear and discontinuous boundary, while aiming to preserve the feasibility of the population, a simple bisection algorithm was also implemented.

While recently, Rosso et al. (2022) a variant of the well-known swarm-based algorithm, the Particle Swarm Optimization (PSO), is developed to solve constrained problems with a different approach to the classical penalty function technique. State-of-art improvements and suggestions are also adopted in the current implementation (inertia weight, neighbourhood). Furthermore, a new local search operator has been implemented to help localize the feasible region in challenging optimization problems. This operator is based on hybridization with another milestone meta-heuristic algorithm, the Evolutionary Strategy (ES). The self-adaptive variant has been adopted because of its advantage of not requiring any other arbitrary parameter to be tuned.

3 Constraint handling technique formulation

In this section, four of the most commonly employed CHTs identified in the literature are described, specifically: (i) penalty methods; (ii) feasibility rules; (iii) ε -constrained method; and (iv) stochastic ranking, while a brief description of methods based on ensemble of CHTs (i)–(iv) is also provided. Furthermore, eight novel variants of the feasibility rules and ε -constrained CHTs are introduced. The performance of these novel variants is evaluated later in Sect. 5, based upon 20 mathematical and engineering COPs, and compared to four existing variants of CHTs (i)–(iv), specifically: (i) adaptive penalty by Kawachi et al. (2019) (ii) feasibility rules by Deb (2000); (iii) improved ε -constrained by Fan et al. (2018); and (iv) stochastic ranking by Runarsson and Yao (2000), which are also described herein.

3.1 Penalty methods

In penalty methods the constrained problem is converted into an unconstrained one with the addition of a penalty term to the objective function. The penalty term depends on the level of violation of the constraint functions, as well as penalty factors whose value can vary throughout the optimization process. The penalized objective function can be expressed in the following general form (Coello and Montes 2002):

$$F(x) = f(x) \pm \left[\sum_{j=1}^J r_j \cdot g_j(x) + \sum_{k=1}^K c_k \cdot h_k(x) \right], \quad (2)$$

where $F(x)$ is the penalized objective function (also called fitness function), $g_j(x)$ and $h_k(x)$ are the inequality and equality constraints, respectively, and r_j and c_k are positive weight coefficients called penalty factors. Converting the equality into inequality constraints, as previously discussed in Sect. 1, Eq. (2) can be rewritten as:

$$F(x) = f(x) \pm \sum_{j=1}^m r_j \cdot g_j(x). \quad (3)$$

Kawachi et al. (2019) proposed an adaptive procedure for calculating the penalty factor during the evolution process, which effectively deals with issues of over- or under- penalization that potentially led to divergence of the search process from the optimum. More specifically, a fitness function is formulated:

$$F(x) = f(x) + PF \cdot \bar{v}(x), \quad (4)$$

where $\bar{v}(x)$ is the mean constraint violation and PF is the penalty factor, and subsequently three steps are implemented. First the penalty factor candidates (PFCs) are calculated by comparing two individuals as follows:

$$PFC_{k,l} = -\frac{f(x_k) - f(x_l)}{\bar{v}(x_k) - \bar{v}(x_l)}, \quad (5)$$

where subscripts k and l denote two distinct individuals; PFC is calculated for all possible combinations in a given population. In the second step, the penalty factor (PF) is defined, as follows: if the percentage of negative $PFCs$ exceeds 50%, the PF remains the same with the one of the previous generations; otherwise PF is calculated as the average value of the positive $PFCs$. In the third step PF is updated. If the proportion of feasible individuals r_f in the population exceeds the p_{feas} , then the penalty factor of the next generation is obtained as follows:

$$PF_{G+1} = p_{rate} \cdot PF_G, \quad (6)$$

where $p_{rate} \in [0, 1]$ and p_{feas} are user defined parameters.

3.2 Feasibility rules

In penalty methods, different values of the penalty factor need to be tested to establish the best option, since inappropriate values can cause search process divergence from the area of the optimum solution. To overcome this shortcoming Deb (2000) proposed a new technique based upon a tournament selection operator for comparing two solutions, in accordance with the following criteria: (i) any feasible solution is preferred to any infeasible solution; (ii) among two feasible solutions, the one associated with better objective function value is preferred; and (iii) among two infeasible solutions, the one associated with smaller constraint violation value is preferred.

3.2.1 Original feasibility rules technique

According to the original feasibility rules technique, the fitness function is defined as follows:

$$F(x) = \begin{cases} f(x) & \text{if } g_j(x) \leq 0 \quad \forall j = 1, 2, \dots, m \\ f_{max} + \sum_{j=1}^m \max\{0, g_j(x)\} & \text{otherwise} \end{cases}, \quad (7)$$

where f_{max} is the objective function value for the worst feasible solution of the current population. If no feasible solution exists in a population, f_{max} is set to zero. According to this technique, all constraint violations are summed, and this sum is compared as a single value. Therefore, in case of infeasible solutions, these are compared based solely on the level of constraint violation.

3.2.2 New variants of the feasibility rules technique

In this section four new variants of the feasibility rules technique are presented. Specifically, the comparison rules described above for the original technique have been modified to bias the search direction towards better feasible solutions, instead of simply feasible ones, to achieve a global exploration of the design space. The formulation of the proposed variants relies on the premise that a small constraint violation should not be the sole criterion of selecting one individual over another, as is the case in existing techniques.

Specifically, when comparing two infeasible individuals, the objective function value and the number of violated constraint functions should also be considered, alongside the maximum constraint violation value, and the product of these three entities shall determine the final winner: if an individual is associated with a lower level of constraint violation than another, but the objective function value of the first is much larger compared to the second, maintaining the first one in the next generation might direct the search process to a local optimum. Of course, if the second individual corresponds to a very large level of constraint violation, it will be penalized with a much larger penalization value. To avoid very small objective function values that could misdirect the search process, the objective function value of the best feasible individual found so far is used as the lower bound. Analogously, when comparing one feasible and one infeasible individual, the second one could be chosen based on the product of its objective function value and the level of the constraint violation. If the objective function value of the feasible solution is very bad, then selecting it might direct the search process to a local optimum, and hence the infeasible individual should be preferred. Finally, since it is not sure which one is better between a solution with many violated constraint functions with small level of violation compared to a solution with just one violated constraint function of very large level of violation, the total number of violated constraint functions is also considered together with the maximum level of constraint violation. Taking the above into consideration, the formulation of the new variants is described hereafter. Please note that the red colour value has been changed to italic in Tables 8, 15, 18, 21, 22. Please check and confirm. Confirm

To calculate the fitness function of an infeasible individual, the $p_{violation}$ factor is introduced. This is the normalized maximum constraint violation of the individual, which, depending on the variant, is multiplied with a term related to the relative number of violated constraint functions for a solution over the total number of constraint functions. The $p_{violation}$ factor for each of the four variants proposed is defined as follows:

$$p_{violation} = \left\| \max \left\{ \max \left\{ 0, \bar{g}_j(x) \right\}, \forall j = 1, 2, \dots, m \right\} \right\| > 1, \quad (8a)$$

$$p_{violation} = \|\max\{\max\{0, \bar{g}_j(x)\}, \forall j = 1, 2, \dots, m\}\| \times \left(1 + \frac{n_{constviol}}{n_{const}}\right) > 1, \tag{8b}$$

$$p_{violation} = \|\text{mean}\{\bar{g}_j(x), \forall j = 1, 2, \dots, m | \bar{g}_j(x) > 0\}\| \times \left(1 + \frac{n_{constviol}}{n_{const}}\right) > 1, \tag{8c}$$

$$p_{violation} = \|\text{median}\{\bar{g}_j(x), \forall j = 1, 2, \dots, m | \bar{g}_j(x) > 0\}\| \times \left(1 + \frac{n_{constviol}}{n_{const}}\right) > 1, \tag{8d}$$

where $\bar{g}_j(x)$ is the normalised value of the j^{th} constraint (an example of a normalised constraint is shown below in Eq. 9b), $n_{constviol}$ is the number of violated constraint functions and n_{const} is the total number of constraint functions. The fitness function of an individual is calculated as the product of $p_{violation}$ with the maximum among the objective function value of the individual and the best objective function value of the feasible individuals found so far, as stated in the following expression:

$$g_j(x) : \sigma(x) \leq \sigma_{all}$$

$$\bar{g}_j(x) : \frac{\sigma(x)}{\sigma_{all}} - 1 \leq 0 \tag{9a}$$

$$F(x) = \begin{cases} f(x) & \text{if } \bar{g}_j(x) \leq 0, \forall j = 1, 2, \dots, m \\ \max(f_{bestfeasible}, f(x)) \times p_{violation} & \text{otherwise} \end{cases} \tag{9b}$$

where an example of a constraint function is shown in Eq. (9b), where $\sigma(x)$ is the principal stress developed into a specific structural element and σ_{all} is the corresponding allowable value. On the basis of expression (9b), three criteria are used to compare two solutions: (i) between two feasible solutions, the one associated with better objective function value is preferred, (ii) between two infeasible solutions, the one associated with lower level of constraint violation and lower number of violated constraint functions is preferred; and (iii) between one feasible and one infeasible solution, the selection of the feasible solution depends on its objective function value, the level of constraint violation and the number of violated constraint functions of the infeasible one. In case of a maximization problem the criteria are adapted accordingly.

3.3 ε-constrained method

A modified version of the feasibility rules technique was originally proposed by Takahama and Sakai (2005), with the aim of providing comparison rules on constrained optimization problems during the evolutionary process. According to this technique, the overall constraint violation function $\varphi(x)$ is defined as either the maximum or the sum of all constraints, and is used in conjunction with the ε level comparison $\{< \varepsilon\}$; for any $\varepsilon \geq 0$, an individual (f_1, φ_1) is considered better than another individual (f_2, φ_2)

according to the following comparison rules:

$$(f_1, \varphi_1) <_{\varepsilon} (f_2, \varphi_2) \iff \begin{cases} f_1 < f_2 & \text{if } \varphi_1, \varphi_2 \leq \varepsilon \\ f_1 < f_2 & \text{if } \varphi_1 = \varphi_2 \\ \varphi_1 < \varphi_2 & \text{otherwise} \end{cases} \quad (10)$$

Therefore: (a) if both overall constraint violations $\varphi(x)$ of two individuals are either less than or equal to ε , the one with better objective function value f is preferred; (b) If the overall constraint violations $\varphi(x)$ of two individuals are equal, the one with better objective function value f is preferred; and (c) if either or both constraint violations $\varphi(x)$ of two individuals are larger than ε , the one with lower constraint violation $\varphi(x)$ is preferred. Building upon their previous work, Takahama and Sakai (2010) suggested that the ε level can be usually controlled according to the following expression:

$$\varepsilon(t) = \begin{cases} \varphi_{\theta} & \text{if } t = 0 \\ \varepsilon(0) \left(1 - \frac{t}{T_C}\right)^{c_p} & \text{if } 0 < t < T_C \\ 0 & \text{if } t \geq T_C \end{cases}, \quad (11)$$

where the initial ε level is equal to φ_{θ} , that is the constraint violation of the top θ^{th} individual in the initial population, t is the number of iterations, T_C is the control generation and c_p is a user defined parameter with value greater or equal to 3.

3.3.1 Improved ε -constrained method

In Fan et al. (2018) proposed the following improved ε setting approach, which applies the ε level comparison $\{<_{\varepsilon}\}$ to establish comparison rules during the evolutionary process, and further balance the evolutionary search of the population between feasible and infeasible regions:

$$\varepsilon(k) = \begin{cases} \varphi_{\theta} & \text{if } k = 0 \\ \varepsilon(k - 1) \left(1 - \frac{FEs}{T_C}\right)^{c_p} & \text{if } r_k < \alpha \text{ and } FEs < T_C \\ (1 + \tau)\varphi_{max} & \text{if } r_k \geq \alpha \text{ and } FEs < T_C \\ 0 & \text{if } FEs \geq T_C \end{cases}, \quad (12)$$

where r_k is the proportion of feasible solutions (PFS) in the generation; φ_{θ} is the θ^{th} largest overall constraint violation of all individuals in the initial population (sorted in an array by their overall constraint violation), where $\theta = \gamma \cdot N_{pop}$, N_{pop} being the population size and $\gamma \in [0.2, 0.8]$; φ_{max} is the largest overall constraint violation of all individuals; T_C is the termination function evaluations (FEs), where $T_C \in [0.1maxFEs, 0.8maxFEs]$; and $c_p \in [2, 10]$, $\tau \in [0, 1]$ and $\alpha \in [0, 1]$ are user-defined parameters.

3.3.2 New variants of the ε -Constrained method

Analogously to the new feasibility rules variants introduced previously in Sect. 3.2.2, new variants of the ε -constrained technique are also proposed herein. The ε level comparison $\{<_{\varepsilon}\}$ in the new variants for two individuals 1, 2 is defined via the following four rules:

$$(f_1, p_1) <_{\varepsilon} (f_2, p_2) \Leftrightarrow \begin{cases} f_1 < f_2 & \text{if } p_1, p_2 \leq \varepsilon \\ f_1 < f_2 & \text{if } p_1 = p_2 \\ \max(f_{bf}, f_1) \times p_1 < \max(f_{bf}, f_2) \times p_2 & \text{if } p_1, p_2 \geq \varepsilon \\ f_1 < \max(f_{bf}, f_2) \times p_2 & \text{otherwise} \end{cases} \quad (13)$$

where f_{bf} stands for the objective function value of the best feasible individual found so far, and p_1, p_2 denote the value of the $p_{violation}$ factor calculated for individuals 1 and 2, based upon Eqs. (8a) to (8d).

Therefore: (a) if both maximum constraint violations of two individuals are either less than or equal to ε level, the one associated with better objective function value f is preferred; (b) if the maximum constraint violations of two individuals are equal, the one associated with better objective function value f is preferred; (c) if both maximum constraint violations of two individuals are larger than the ε level, the one associated with better product of constraint violation and objective function value is preferred; and (d) if only one maximum constraint violation of two individuals is larger than the ε level, the other individual is preferred only if its objective function value is better than the product of the constraint violation with the objective function value of the second individual. In line with the new variants of the feasibility rules technique, in addition to the maximum constraint violation value the total number of violated constraint functions is also considered. To control the ε level, the formulation proposed by Fan et al. (2018) is adopted, as given in Eq. (12).

3.4 Stochastic ranking method

A frequently employed CHT that circumvents the shortcomings of penalty techniques is the stochastic ranking method, originally proposed by Runarsson and Yao (2000). Stochastic ranking is based upon comparing all adjacent individuals in a population according to penalty and objective function dominance, to rank them from best to worst.

A probability $P_f \in [0, 1]$ associated with the sole use of the objective function for the adjacent individual comparisons must be defined by the designer to achieve enhanced performance. A value $P_f = 1/2$ means that the probability of a comparison based on the objective function is the same as the probability based on the penalty function. Ranking is achieved via a bubble-sort-type algorithm, where λ individuals are ranked by comparing adjacent individuals in N sweeps (the least λ sweeps) in every generation. If both adjacent individuals are feasible or a randomly generated number u is less than the selected probability P_f , the individuals are compared according to

their objective function values; otherwise, if one or both are infeasible, the comparison is based on the constraint violation value. The algorithm is terminated if no change takes place in the ranking order within a complete sweep.

3.5 Ensemble of constraint handling techniques

It is not always easy to determine which CHT is better for determining the global optimum in a given COP. While a specific CHT might be more suitable than others for a particular stage of the evolution process, it might be inappropriate for another one, and hence different CHTs might be more effective during different stages of the search process. This depends on several factors, such as the multimodality of the problem, the proportion of feasible search region in the whole search region, the adopted algorithm etc. Motivated by these observations, Mallipeddi and Suganthan (2010) proposed a combination of four CHTs (feasibility rules, self-adaptive penalty, ϵ -constraint method, and stochastic ranking), a scheme called ECHT, to examine constrained optimization problems, where each technique handles a different population. Parents and offsprings of the population compete not only among each other, but also with the respective entities of all the other populations. This means that if an offspring generated from a particular population is not selected for the next generation based on this population's specific CHT, it could be selected by other populations that use a different CHT.

4 Optimization computing platform and pity beetle algorithm

The solution of real-world, large-scale optimization problems within the field of structural engineering can only be achieved with a synergy of the following aspects (Melchiorre et al. 2021; Lagaros et al. 2022; Cucuzza et al. 2022): (i) efficient numerical modelling of the physical problem; (ii) reliable numerical optimization algorithms for enhanced structural design; (iii) rationalized modelling of the system uncertainties (in the case of probabilistic design problems); and (iv) exploitation of modern High-Performance Computing (HPC) facilities. The original OCP (Lagaros 2014) and its evolution HP-OCP offer all aforementioned capabilities, including efficient numerical modelling tools and several MOAs with a great variety of embedded features and CHTs, while its implementation enables its straightforward upgrade to incorporate new tools and features.

In Kallioras et al. (2018) introduced the novel nature-inspired pity beetle MOA, a single-objective optimization algorithm for unconstrained problems which has been shown to handle NP-hard optimization problems efficiently. PBA is inspired by the behavior of the six-toothed bark beetle (*pityogenes chalcographus* beetle) when searching for food on the bark of trees and simulates the beetle behavior in three stages: (i) population initialization; (ii) population regeneration; and (iii) location update. In the initialization stage (i), a population consisting of males and females is randomly allocated in the search space via a random sampling technique (RST), which ensures adequate diversification to avoid premature convergences. The male particles act as pioneers in search for the most suitable host, producing pheromone that attracts the

other males and females. In the population regeneration stage (ii), each single particle seeks an improved position in the search space to create its own population. This is achieved via five types of hypervolume selection patterns, namely global-scale, large-scale, mid-scale, neighboring search, and memory consideration search volume patterns, where in the latter the best positions identified are stored in memory. In the location update stage (iii), the position of each mating male and female is updated and the previous positions are eliminated, except for those stored in memory (memory consideration search volume). Collectively, the hypervolume patterns are selected in accordance with the following expression:

$$x_{j,k}^{(g)} = \begin{cases} \text{RST}\left(\left[x_{birth,i}^{(g)} \cdot (1 - f_{nb}), x_{birth,i}^{(g)} \cdot (1 + f_{nb})\right], n, N_{pop}\right), & \text{if } k = 1 \\ \text{else} \begin{cases} \text{RST}\left(x_i^l, x_i^u, n, N_{pop}\right), & \text{if } FE > FE_{un} \\ \text{RST}\left(\left[x_{birth,i}^{(g)} \cdot (1 - f_{ms}), x_{birth,i}^{(g)} \cdot (1 + f_{ms})\right], n, N_{pop}\right), \exists j f(x_{j,k-1}^{(g)}) < f(x_{birth}^{(g)}) \\ \text{else} \begin{cases} \text{RST}\left(\left[x_{birth,i}^{(g)} \cdot (1 - f_{ls}), x_{birth,i}^{(g)} \cdot (1 + f_{ls})\right], n, N_{pop}\right), & \text{if } r < pr \\ MEM, & \text{otherwise} \end{cases} \end{cases} \end{cases} \quad i = 1, 2, \dots, n, j = 1, 2, \dots, N_{pop}, k = 1, 2, \dots, N_{broods}. \tag{14}$$

In expression (14): g refers to the current generation (pursuit step); i signifies the component of the design vector; j denotes the member of the population; k designates the brood (colony); n denotes the number of unknowns; N_{POP} is the population size; N_{broods} is the maximum number of broods (termination criterion); $x_{birth,i}^{(g)}$ is the i^{th} element of the birth position/solution vector; $x_{j,k}^{(g)}$ is the new position vector found in population k ; superscripts l, u denote the lower and upper bound; while the algorithmic parameters of PBA are: the population of pioneer particles N_{pop} ; the neighboring factor f_{nb} ; the fine tuning factor f_{in} , which is used to define the area size where a very narrow local search is performed, in case a better position is identified; the mid-scale factor f_{ms} ; the large-scale factor f_{ls} ; the probability of choosing large-scale search or memory consideration pr ; and the function evaluations multiplication factor f_{FE} , which is used to define the maximum number of unsuccessful function evaluations FE_{un} with reference to the total function evaluations $FE_{un} = FE_{total} \cdot f_{FE}$; and MEM denotes the memory of size N_{pop} , where the best solutions found by the algorithm up until the current step are stored in memory. A detailed description of PBA can be found in Kallioras et al. (2018).

5 Numerical examples

In this section, the performance of the 8 novel feasibility rules and ϵ -constrained CHT variants proposed in Sect. 3 is evaluated and compared against 4 well-known CHT variants from the literature: (i) adaptive penalty by Kawachi et al. (2019) (ii) feasibility rules by Deb (2000); (iii) improved ϵ -constrained by Fan et al. (2018); and (iv) stochastic ranking by Runarsson and Yao (2000). Consideration is given to 20 COPs previously investigated by other authors, encompassing: (i) 14 mathematical COPs,

including a 2D problem investigated by Deb (2000) and 13 problems investigated by Runarsson and Yao (2000) (labelled G1–G13); and (ii) 6 engineering COPs, including a welded beam, pressure vessel and tension–compression string design investigated by Tsipianitis and Tsompanakis (2020), as well as 2D and 3D truss design examples investigated by Farshi and Alinia-ziazi (2010).

For each numerical example and each employed CHT, 20 independent optimization computations are performed (20 COPs \times 12 CHTs \times 20 computations = 4800 total computations), to establish an adequate sample for a statistical analysis of the obtained results. In all computations the PBA (Kallioras et al. 2018) implemented in the MATLAB variant of HP-OCF is employed (Sect. 4), using identical parameters, termination criteria and bound-constraint treatment (out of bounds solution vectors corrected via projection to the closest bound), which eliminates the influence of the PBA specifications, thus enabling a common basis for CHT comparison to be established. The adopted PBA parameters are: $N_{pop} = 30$, $f_{nb} = 0.08$, $f_{in} = 0.015$, $f_{ms} = 0.90$, $f_{ls} = 100$, $f_{FE} = 0.25$ and $pr = 0.20$ (Kallioras et al. 2018), as well as $FE_{un} = 20,000$. The specifications of the employed CHT variants are given in Table 2. All computations were conducted using MATLAB[®] version R2020b, on an x64 PC with 14-Core Intel[®] Xeon[®] processor with 128 GB RAM memory.

The results obtained for each numerical example using the 12 CHT variants are given in Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 and 23, which include the best, median, worst, and mean optimal solutions achieved amongst the 20 respective computations in each case, as well as the coefficient of variation (CoV) as a standardized dispersion measure. The CHTs achieving the best performance, in respect of objective function value and CoV in each numerical test, are highlighted in bold in Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,

Table 2 Specifications of constrained handling techniques employed in numerical examples

ID	Acronym	Description	Expression
1	FeasRulesOR	Original version of the feasibility rules	Equation (7)
2	FeasRulesNEW1	Variant 1 of the feasibility rules	Equations (8a) and (9)
3	AdPenalty	Adaptive penalty technique	Eqs. (4), (5) and (6)
4	EconstOR	Original version of the ϵ -constraint technique	Equations (10) and (12)
5	EconstNEW1	Variant 1 of the ϵ -constraint technique	Equations (8a), (12) and (13)
6	StochRanking	Stochastic ranking technique	See corresponding section
7	FeasRulesNEW2	Variant 2 of the feasibility rules	Equations (8b) and (9)
8	FeasRulesNEW3	Variant 3 of the feasibility rules	Equations (8c) and (9)
9	FeasRulesNEW4	Variant 4 of the feasibility rules	Equations (8d) and (9)
10	EconstNEW2	Variant 2 of the ϵ -constraint technique	Equations (8b), (12) and (13)
11	EconstNEW3	Variant 3 of the ϵ -constraint technique	Equations (8c), (12) and (13)
12	EconstNEW4	Variant 4 of the ϵ -constraint technique	Equations (8d), (12) and (13)

Table 3 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem by Deb

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	13.6405	13.6438	13.7633	13.6685	0.389%
2	13.6029	13.6319	13.6590	13.6297	0.187%
3	13.5919	13.6680	13.7080	13.6480	0.357%
4	13.5924	13.6007	13.6721	13.6130	0.245%
5	13.6016	13.6445	13.6766	13.6390	0.205%
6	15.8271	17.3333	22.8357	18.8444	17.828%
7	13.6043	13.6218	13.7207	13.6424	0.350%
8	13.6104	13.6299	13.6809	13.6403	0.201%
9	13.5912	13.6716	13.7148	13.6549	0.393%
10	13.5944	13.6112	13.6397	13.6137	0.142%
11	13.6163	13.6476	13.7437	13.6736	0.388%
12	13.6052	13.6329	13.6598	13.6348	0.151%

Table 4 Specifications of mathematical constrained optimization problems G1-G13

Numerical Example	Optimization Type	Objective Function Type	Objective Function Value	Variables	LE	LI	NE	NI
G1	Min	Quadratic	- 15	13	0	9	0	0
G2	MAX	Nonlinear	0.803619	20	0	1	0	1
G3	MAX	Nonlinear	1	20	0	0	1	0
G4	Min	Quadratic	- 30,665.539	5	0	0	0	6
G5	Min	Cubic	5126.4981	4	0	2	3	0
G6	Min	Cubic	- 6961.81388	2	0	0	0	2
G7	Min	Quadratic	24.3062091	10	0	3	0	5
G8	MAX	Nonlinear	0.095825	2	0	0	0	2
G9	Min	Polynomial	680.6300573	7	0	0	0	4
G10	Min	Linear	7049.3307	8	0	3	0	3
G11	Min	Quadratic	0.75	2	0	0	1	0
G12	Min	Quadratic	1	3	0	0	0	9 ³
G13	Min	Nonlinear	0.0539498	5	0	0	3	0

Table 5 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G1

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	- 14.03	- 13.96	- 6.51	- 11.21	34.08%
2	- 14.76	- 8.07	- 6.96	- 10.29	38.14%
3	- 14.72	- 14.40	- 6.13	- 11.25	40.04%
4	- 14.87	- 14.13	- 6.13	- 11.30	40.00%
5	- 14.38	- 13.87	- 6.68	- 12.54	26.18%
6	- 14.75	- 14.15	- 13.16	- 13.92	4.81%
7	- 14.41	- 7.04	- 6.27	- 9.68	44.39%
8	- 14.52	- 13.73	- 13.41	- 13.82	3.25%
9	- 14.99	- 14.01	- 13.56	- 14.15	3.96%
10	- 14.86	- 14.00	- 6.41	- 11.23	38.14%
11	- 14.94	- 14.28	- 13.41	- 14.08	4.67%
12	- 14.22	- 13.93	- 13.79	- 13.97	1.36%

Table 6 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G2

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	0.7711604	0.7350424	0.7260033	0.7426673	2.48%
2	0.7930772	0.7403387	0.4855716	0.6514598	23.16%
3	0.7614887	0.5777608	0.4661263	0.5817679	19.41%
4	0.5022935	0.4659332	0.4623640	0.4783193	4.03%
5	0.4816939	0.4573892	0.4531349	0.4644524	2.84%
6	0.7511018	0.7411717	0.7229049	0.7371014	1.74%
7	0.7763752	0.4745132	0.4571342	0.5890534	28.34%
8	0.7774812	0.7344782	0.4568700	0.6531579	22.15%
9	0.4851802	0.4676234	0.4603978	0.4721634	2.45%
10	0.7992425	0.7491075	0.4827461	0.6637055	23.16%
11	0.7920154	0.7334940	0.4514678	0.6918536	19.82%
12	0.7911422	0.4625053	0.4481425	0.5290209	27.90%

Table 7 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G3

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	0.975806	0.957341	0.934324	0.955949	1.55%
2	0.985784	0.955447	0.907347	0.949690	3.02%
3	0.985461	0.941339	0.932888	0.951327	2.22%
4	0.971099	0.945975	0.902593	0.938104	3.16%
5	0.966954	0.937744	0.895094	0.927956	3.19%
6	0.990351	0.959357	0.836282	0.931881	7.23%
7	0.956951	0.927027	0.897288	0.928813	2.66%
8	0.938294	0.907438	0.893265	0.911338	1.84%
9	0.983752	0.928372	0.904847	0.936557	3.42%
10	0.995233	0.952833	0.894044	0.953569	4.30%
11	0.972879	0.923372	0.898509	0.934129	3.31%
12	0.946160	0.936331	0.929124	0.937168	0.66%

19, 20, 21, 22 and 23 and summarized in Table 24. Cases where infeasible solutions were derived are marked in italic in Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 and 23. In Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 and 23 the *Best* out of the optimum solution achieved through the 20 independent optimization runs along with the corresponding *Median*, *Worst*, *Mean* and the coefficient of variation (*CoV*) that is considered a standardized measure for the dispersion, are used for assessing the performance of each constraint handling technique for specific test example. The techniques with the better performance in each problem are summarized in Table 24.

5.1 Mathematical constrained optimization problems

The first mathematical COP is a 2D minimization problem originally introduced by Deb (2000), which is formulated as follows:

$$\begin{aligned}
 & \text{Minimize } f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2, \\
 & \text{subject to :} \\
 & g_1(x) = (x_1 - 0.05)^2 + (x_2 - 2.5)^2 - 4.84 \leq 0, \\
 & g_2(x) = 4.84 - x_1^2 - (x_2 - 2.5)^2 \leq 0, \\
 & 0 \leq x_1, x_2 \leq 6.
 \end{aligned} \tag{15}$$

The reported objective function value in Deb (2000) is equal to 13.59658. The results obtained using the PBA and the various CHTs are given in Table 3. The best

Table 8 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G4 (solutions denoted in *italic* correspond to infeasible designs)

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	– 30,513.242	– 30,299.343	– 30,046.170	– 30,297.569	0.56%
2	– 30,407.490	– 30,195.412	– 30,099.417	– 30,229.828	0.38%
3	– 30,541.540	– 30,175.502	– 30,146.467	– 30,282.546	0.57%
4	– 30,266.240	– 30,144.791	– 30,014.593	– 30,155.414	0.33%
5	– 30,388.244	– 30,290.855	– 30,241.275	– 30,318.903	0.22%
6	– <i>31,504.913</i>	– 29,339.393	– 28,762.364	– 29,846.753	4.02%
7	– 30,540.597	– 30,210.981	– 30,182.475	– 30,287.673	0.49%
8	– 30,421.757	– 30,242.244	– 30,153.785	– 30,270.515	0.34%
9	– 30,227.128	– 30,107.508	– 29,967.899	– 30,104.948	0.35%
10	– 30,371.951	– 30,304.734	– 30,190.117	– 30,288.479	0.23%
11	– 30,180.255	– 30,135.559	– 30,001.176	– 30,099.464	0.26%
12	– 30,295.772	– 30,136.911	– 30,123.318	– 30,185.524	0.25%

performance was achieved by *AdPenalty*, *EconstOR*, *FeasRulesNEW4* and *EconstNEW2* CHTs with an optimized objective function value lower than 13.60, while *EconstNEW2* achieved the lowest CoV value of 0.142%

Subsequently, consideration is given to 13 mathematical COPs labelled G1–G13, originally introduced by Runarsson and Yao (2000). The problem specifications are given in Table 4, where LE, LI, NE and NI respectively denote the number of Linear Equality, Linear Inequality, Nonlinear Equality and Nonlinear Inequality constraint functions. The results obtained using the PBA and the various CHTs are given in Tables 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17.

G1 is a 13-dimensional minimization problem with 9 constraint functions and reference objective function value equal to – 15.0 (Runarsson and Yao 2000). The respective results are given in Table 5. All CHTs produced optimized objective function values below – 14.0 and the highest performance was achieved by *FeasRulesNEW4* with – 14.99. The lowest CoV value of 1.36% was achieved by *EconstNEW4*.

Table 9 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G5

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	5177.734	5346.547	5653.663	5397.877	4.04%
2	5215.865	5553.208	5658.655	5471.029	3.23%
3	5298.059	5561.201	5753.248	5550.329	3.26%
4	5179.488	5364.924	5599.945	5344.883	3.11%
5	5292.990	5320.135	5423.499	5344.439	0.98%
6	2108.137	5521.316	6139.343	5032.023	33.27%
7	5195.300	5512.515	5675.418	5482.298	3.42%
8	5140.124	5312.207	5723.401	5377.013	4.69%
9	5368.789	5469.188	5693.304	5534.914	2.66%
10	5192.350	5358.056	5698.489	5417.474	3.94%
11	5127.392	5331.650	5378.559	5290.603	2.01%
12	5237.019	5506.964	5724.377	5447.294	3.71%

Table 10 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G6

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	- 6790.586	- 6358.391	- 6159.697	- 6382.404	3.85%
2	- 6912.992	- 6420.708	- 4858.393	- 6208.230	12.81%
3	- 6412.182	- 5862.079	- 4240.101	- 5414.184	19.34%
4	- 6667.991	- 5914.872	- 5558.692	- 6139.985	8.09%
5	- 6816.872	- 6499.084	- 4567.834	- 5834.570	18.36%
6	- 6537.467	- 6071.105	4710.211	- 4027.661	121.41%
7	- 6365.516	- 4712.400	- 2944.879	- 4773.019	31.70%
8	- 6479.756	- 4986.613	- 2863.977	- 5021.442	30.21%
9	- 6597.254	- 6397.199	- 3558.333	- 5837.619	21.95%
10	- 6829.290	- 6304.736	- 6217.506	- 6377.708	4.01%
11	- 6711.243	- 6306.255	- 4203.241	- 5631.175	21.73%
12	- 6803.016	- 6669.477	- 4679.405	- 6312.298	14.49%

Table 11 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G7

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	24.335	25.473	26.848	25.749	4.23%
2	24.774	25.902	26.680	25.688	2.89%
3	24.859	25.672	26.537	25.743	2.94%
4	24.718	26.237	26.756	25.836	3.35%
5	24.524	25.053	26.408	25.180	2.91%
6	26.730	28.032	29.099	27.934	3.63%
7	25.164	26.364	26.575	26.119	2.16%
8	24.355	26.187	27.308	25.917	4.77%
9	25.156	26.379	26.790	26.263	2.53%
10	24.395	25.960	26.959	25.911	3.65%
11	24.876	26.201	27.273	26.160	4.08%
12	24.559	25.396	25.908	25.377	2.03%

Table 12 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G8

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	0.094446	0.082134	0.079666	0.084812	7.06%
2	0.090791	0.086852	0.061727	0.082233	14.69%
3	0.095308	0.092914	0.078204	0.090294	7.73%
4	0.095792	0.095702	0.095505	0.095648	0.13%
5	0.088702	0.077539	0.064085	0.078165	11.86%
6	0.092216	0.081623	0.079876	0.085110	6.68%
7	0.093141	0.092203	0.084332	0.089818	4.61%
8	0.088445	0.084444	0.076421	0.082722	6.64%
9	0.094060	0.084896	0.083960	0.087940	5.59%
10	0.090096	0.075877	0.074068	0.080064	9.32%
11	0.091973	0.091749	0.081898	0.088652	5.18%
12	0.095506	0.088928	0.064165	0.083105	15.59%

Table 13 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G9

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	897.707	920.974	1159.158	963.366	11.41%
2	1000.375	1191.478	1194.215	1136.127	7.51%
3	872.839	891.833	1196.815	952.178	14.49%
4	894.120	899.778	921.734	906.594	1.50%
5	923.739	981.411	1158.647	1000.063	9.19%
6	740.834	758.508	1282.632	914.006	26.54%
7	955.942	1002.580	1045.615	995.212	3.85%
8	738.137	1399.981	1549.860	1199.989	32.32%
9	1155.802	1190.193	1470.834	1279.561	11.75%
10	862.965	1092.886	1187.899	1060.987	13.29%
11	807.131	899.927	1145.067	944.924	13.67%
12	743.261	941.784	1188.560	949.794	21.50%

Table 14 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G10

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	7201.763	7638.056	7864.345	7560.307	3.32%
2	7375.896	7604.305	7833.286	7580.116	2.27%
3	7144.402	7149.005	7459.088	7268.024	2.29%
4	7104.735	7607.826	7876.609	7555.498	4.50%
5	7058.463	7714.851	7886.231	7528.165	4.96%
6	8037.758	12,483.940	16,783.772	12,038.132	32.67%
7	7705.551	7858.799	7914.563	7839.422	1.07%
8	7080.111	7207.520	7772.763	7292.102	3.88%
9	7341.473	7558.836	7802.294	7591.445	2.39%
10	7209.965	7419.512	7615.370	7406.549	2.34%
11	7230.334	7614.661	7879.947	7556.595	3.87%
12	7197.730	7391.895	7818.409	7494.428	3.43%

Table 15 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G11 (solutions denoted in *italic* correspond to infeasible designs)

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	0.765	0.843	1.000	0.875	13.53%
2	0.763	0.829	0.932	0.831	8.35%
3	0.753	0.759	0.806	0.774	3.11%
4	0.750	0.804	0.924	0.816	9.03%
5	0.751	0.780	0.818	0.783	3.76%
6	<i>0.645</i>	0.781	1.430	0.887	36.05%
7	0.752	0.763	0.997	0.809	13.05%
8	0.775	0.812	0.998	0.845	10.41%
9	0.766	0.800	0.837	0.802	3.48%
10	0.750	0.764	0.836	0.785	4.75%
11	0.761	0.827	0.989	0.870	12.65%
12	0.756	0.794	1.000	0.863	14.06%

Table 16 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G12

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	1.00000	1.00002	1.00005	1.00002	0.00%
2	1.00006	1.00012	1.00014	1.00011	0.00%
3	1.00000	1.00000	1.00005	1.00002	0.00%
4	1.00000	1.00002	1.00011	1.00004	0.00%
5	1.00005	1.00006	1.00013	1.00008	0.00%
6	1.00614	1.02384	1.04917	1.02770	1.75%
7	1.00003	1.00624	1.00726	1.00399	0.36%
8	1.00003	1.00010	1.00045	1.00018	0.02%
9	1.00001	1.00008	1.00022	1.00008	0.01%
10	1.00000	1.00004	1.00022	1.00009	0.01%
11	1.00008	1.00031	1.00778	1.00173	0.34%
12	1.00001	1.00009	1.00054	1.00022	0.02%

Table 17 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in optimization problem G13

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	0.05403	0.05551	0.05906	0.05571	3.57%
2	0.05484	0.05539	0.06006	0.05630	3.79%
3	0.05460	0.05808	0.05923	0.05733	3.26%
4	0.05541	0.05603	0.05861	0.05639	2.30%
5	0.05402	0.05635	0.05891	0.05667	3.62%
6	0.05804	0.06149	0.06396	0.06141	3.50%
7	0.05458	0.05793	0.06044	0.05795	3.79%
8	0.05569	0.05708	0.05992	0.05758	3.19%
9	0.05454	0.05936	0.05986	0.05832	3.72%
10	0.05534	0.05632	0.05792	0.05663	1.78%
11	0.05462	0.05572	0.05796	0.05587	2.34%
12	0.05400	0.05499	0.05917	0.05622	4.36%

Table 18 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in the welded beam design optimization problem (solutions denoted in italic correspond to infeasible designs)

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	1.819183	1.877287	2.080079	1.896935	5.61%
2	1.783706	1.805245	1.867979	1.817916	2.05%
3	1.821214	1.832601	1.886860	1.845314	1.47%
4	1.783271	1.812088	1.846900	1.814719	1.33%
5	1.768195	1.843518	1.896582	1.841706	2.58%
6	<i>0.789035</i>	1.733331	2.624647	1.741458	42.68%
7	1.758677	1.776146	1.859217	1.795085	2.41%
8	2.693128	3.282202	3.874547	3.329804	14.90%
9	1.734493	1.897411	2.249491	1.946973	9.68%
10	1.802315	1.816971	1.865189	1.827903	1.35%
11	1.729478	1.780118	2.473322	1.909538	16.55%
12	1.751654	1.799929	2.155062	1.876898	8.93%

Table 19 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in the pressure vessel design optimization problem

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	6368.00	6810.20	7618.46	6950.99	8.13%
2	5987.72	6363.00	6488.27	6312.30	3.16%
3	5767.50	6311.44	6913.15	6269.59	6.84%
4	6017.10	6165.67	7100.19	6398.22	7.25%
5	6259.06	6509.27	7578.77	6679.68	7.79%
6	5951.17	6378.09	6731.42	6413.13	4.86%
7	6022.30	6546.09	6991.38	6560.92	5.56%
8	5964.41	6250.48	7222.21	6458.31	7.92%
9	6299.75	10,956.91	12,655.53	9578.73	30.70%
10	6681.82	6733.53	9823.63	7586.19	18.09%
11	5901.23	6484.85	8905.69	6840.02	17.55%
12	5953.96	6869.58	9691.62	7122.72	21.33%

Table 20 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in the tension–compression string design optimization problem

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	0.0127465	0.0132460	0.0134696	0.0131711	2.47%
2	0.0126911	0.0135908	0.0142398	0.0135285	4.21%
3	0.0127114	0.0129065	0.0140891	0.0132040	4.59%
4	0.0130122	0.0132534	0.0140961	0.0134624	3.31%
5	0.0129895	0.0133000	0.0141234	0.0135039	4.09%
6	0.0129869	0.0132778	0.0146449	0.0136458	5.23%
7	0.0130214	0.0138538	0.0141068	0.0137407	3.08%
8	0.0128321	0.0133278	0.0141465	0.0134180	3.55%
9	0.0129781	0.0132390	0.0136484	0.0132572	1.91%
10	0.0129213	0.0133623	0.0140796	0.0134175	3.65%
11	0.0127440	0.0134109	0.0137481	0.0132890	2.88%
12	0.0132850	0.0136333	0.0138492	0.0136283	1.69%

Table 21 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in the 2D 10-bar truss design optimization problem (solutions denoted in italic correspond to infeasible designs)

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	5402.606	5645.476	5826.191	5626.784	2.69%
2	5442.038	5723.340	5766.712	5643.266	2.62%
3	5293.971	5579.619	6289.305	5701.531	6.87%
4	5619.414	5986.972	6137.717	5883.129	4.05%
5	5674.229	5879.283	6048.491	5875.668	2.27%
6	<i>2275.305</i>	3854.001	5895.347	3829.106	37.97%
7	5324.794	5714.612	5886.631	5639.842	3.84%
8	5527.953	5721.189	5878.250	5738.699	2.44%
9	5440.772	5610.494	5861.193	5613.476	2.92%
10	5437.570	5688.695	5963.976	5658.574	3.63%
11	5458.437	5689.077	5896.616	5659.801	3.01%
12	5385.125	5512.638	6068.430	5594.182	4.90%

Table 22 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in the 3D 25-bar truss design optimization problem (solutions denoted in italic correspond to infeasible designs)

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	558.305	565.372	576.959	566.204	1.30%
2	555.235	566.208	576.083	565.685	1.54%
3	557.540	567.227	571.198	564.360	1.09%
4	570.237	575.056	591.023	578.260	1.49%
5	557.583	577.849	586.473	573.751	1.94%
6	<i>240.610</i>	264.651	597.525	387.299	46.66%
7	563.104	569.576	577.552	570.890	0.98%
8	566.151	575.179	584.689	574.374	1.34%
9	559.576	566.308	573.174	566.697	0.89%
10	560.904	572.653	576.329	569.545	1.13%
11	557.275	565.446	576.859	566.122	1.33%
12	555.043	561.412	566.625	562.013	0.82%

Table 23 Best, median, worst, and mean objective function values and coefficient of variation obtained for the various constraint handling techniques in the 3D 72-bar truss design optimization problem

Constraint handling technique	Best solution	Median solution	Worst solution	Mean solution	CoV(%)
1	403.107	411.682	429.065	414.864	2.47%
2	401.547	423.876	434.365	422.422	2.97%
3	419.607	424.087	425.538	423.308	0.54%
4	445.125	452.496	464.080	454.607	1.86%
5	409.065	435.853	439.018	429.530	2.97%
6	512.596	691.042	832.441	673.067	17.18%
7	399.669	422.896	438.794	419.232	3.65%
8	400.619	415.966	431.441	415.780	2.83%
9	400.818	417.094	435.493	418.635	3.05%
10	398.509	406.776	416.210	407.859	1.78%
11	402.448	420.500	448.062	422.559	3.89%
12	411.693	418.010	425.763	418.154	1.31%

G2 is a 20-dimensional maximization problem with 2 constraint functions and reference objective function value equal to 803,619 (Runarsson and Yao 2000). The respective results are given in Table 6. Almost half of the CHTs produced optimized objective function values greater than 0.75 and the highest performance was achieved by *EconstNEW2* with 0.7992425. The lowest CoV value of 1.74% was achieved by *StochRanking*.

G3 is a 20-dimensional maximization problem with 1 constraint function and reference objective function value equal to 1.0 (Runarsson and Yao 2000). The respective results are given in Table 7. Almost all CHTs produced optimized objective function values greater than 0.95 and the highest performance was achieved by *EconstNEW2* with 0.995233. The lowest CoV value of 0.66% was achieved by *EconstNEW4*.

G4 is a 5-dimensional minimization problem with 6 constraint functions and reference objective function value equal to $-30,665.539$ (Runarsson and Yao 2000). The respective results are given in Table 8. Almost all CHTs produced optimized objective function values lower than $-30,000$ and the highest performance was achieved by *AdPenalty* with $-30,541.540$. Nearly all techniques achieved CoV values less than 1.0%, with the lowest corresponding to *EconstNEW1*, *StochRanking* failed to converge.

G5 is a 4-dimensional minimization problem with 5 constraint functions and reference objective function value equal to 5126.4981 (Runarsson and Yao 2000). The respective results are given in Table 9. Almost all CHTs produced optimized objective function values lower than 5300 and the highest performance was achieved by *EconstNEW3* with 5127.392. The lowest CoV value of 0.98% was achieved by *EconstNEW1*.

G6 is a 2-dimensional minimization problem with 2 constraint functions and reference objective function value equal to -6961.81388 (Runarsson and Yao 2000). The

Table 24 Synopsis of best-performing constraint handling techniques in numerical examples

Numerical example	Objective function value	CoV(%)
Deb	AdPenalty, EconstOR, FeasRulesNEW4, EconstNEW2	EconstNEW2
G1	FeasRulesNEW4	EconstNEW4
G2	EconstNEW2	StochRanking
G3	EconstNEW2	EconstNEW4
G4	AdPenalty	EconstNEW1 (almost all less than 1%)
G5	EconstNEW3	EconstNEW1
G6	FeasRulesNEW1	FeasRulesOR, EconstNEW2
G7	FeasRulesOR	EconstNEW4 (almost all less than 4%)
G8	EconstOR	EconstOR
G9	FeasRulesNEW3	EconstOR
G10	EconstNEW1	FeasRulesNEW2
G11	AdPenalty, EconstOR, EconstNEW1, FeasRulesNEW2, EconstNEW2, EconstNEW4	AdPenalty
G12	Almost all	Almost all
G13	FeasRulesOR, EconstNEW1, EconstNEW4	EconstNEW2
Welded beam design	EconstNEW3	AdPenalty, EconstOR, EconstNEW2
Pressure vessel design	FeasRulesNEW1, AdPenalty, StochRanking, FeasRulesNEW3, EconstNEW3, EconstNEW4	FeasRulesNEW1
Tension–compression string design	FeasRulesNEW1	FeasRulesNEW4, EconstNEW4
10-bar truss design	AdPenalty	FeasRulesOR, FeasRulesNEW1, EconstNEW1, FeasRulesNEW3, FeasRulesNEW4
25-bar truss design	FeasRulesOR, FeasRulesNEW1, AdPenalty, EconstNEW1, FeasRulesNEW4, EconstNEW3, EconstNEW4	FeasRulesNEW2, FeasRulesNEW4, EconstNEW4
72-bar truss design	FeasRulesNEW2, EconstNEW2	AdPenalty

respective results are given in Table 10. Almost half CHTs produced optimized objective function values lower than -6700 and the highest performance was achieved by *FeasRulesNEW1* with -6912.992 . The lowest CoV values of 3.85% and 4.01% were respectively achieved by *FeasRulesOR* and *EconstNEW2*.

G7 is a 10-dimensional minimization problem with 8 constraint functions and reference objective function value equal to 24.3062091 (Runarsson and Yao 2000). The respective results are given in Table 11. Almost all CHTs produced optimized objective function values lower than 25 and the highest performance was achieved by *FeasRulesOR* with 24.335. Nearly all techniques achieved CoV values less than 4.0%; the lowest CoV values of 2.03% and 2.16% were respectively achieved by *EconstNEW4* and *FeasRulesNEW2*.

G8 is a 2-dimensional maximization problem with 2 constraint functions and reference objective function value equal to 0.095825 (Runarsson and Yao 2000). The respective results are given in Table 12. Almost all CHTs produced optimized objective function values greater than 0.090 and the highest performance was achieved by *EconstOR* with 24.335. The lowest CoV values of 0.13% was also achieved by *EconstOR*.

G9 is a 7-dimensional minimization problem with 4 constraint functions and reference objective function value equal to 680.6300573 (Runarsson and Yao 2000). The respective results are given in Table 13. Almost all CHTs produced optimized objective function values greater than 750 and the highest performance was achieved by *FeasRulesNEW3* with 738.137. The lowest CoV value of 1.50% was achieved by *EconstOR*.

G10 is a 8-dimensional minimization problem with 6 constraint functions and reference objective function value equal to 7049.3307 (Runarsson and Yao 2000). The respective results are given in Table 14. Almost all CHTs produced optimized objective function values lower than 7400 and the highest performance was achieved by *EconstNEW1* with 7058.463. The lowest CoV value of 1.07% was achieved by *FeasRulesNEW2*.

G11 is a 2-dimensional minimization problem with 1 constraint functions and reference objective function value equal to 0.75 (Runarsson and Yao 2000). The respective results are given in Table 15. Almost all CHTs produced optimized objective function values lower than 0.78 and the highest performance was achieved by *AdPenalty*, *EconstOR*, *EconstNEW1*, *FeasRulesNEW2*, *EconstNEW2* and *EconstNEW4* with values very close to 0.75. The lowest CoV value of 3.11% was achieved by *AdPenalty*. *StochRanking* failed to converge.

G12 is a 3-dimensional minimization problem with 729 constraint functions and reference objective function value equal to 1 (Runarsson and Yao 2000). The respective results are given in Table 16. All CHTs produced the reference objective function value, with most of them being associated with an extremely low CoV below 0.02%.

G13 is a 5-dimensional minimization problem with 3 constraint functions and reference objective function value equal to 0.0539498 (Runarsson and Yao 2000). All CHTs produced optimized objective function values lower than 0.056 and the highest performance was achieved by *FeasRulesOR*, *EconstNEW1* and *EconstNEW4* with values very close to 0.054. The lowest CoV value of 1.78% was achieved by *EconstNEW2*.

5.2 Engineering constrained optimization problems

5.2.1 Welded beam design

This COP concerns the optimal design of a welded beam via minimization of the fabrication cost, which depends on four design variables: weld thickness x_1 , weld length x_2 , beam height x_3 , and beam width x_4 . The problem is formulated as follows:

$$\begin{aligned}
 & \text{Minimize } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2), \\
 & \text{subject :} \\
 & g_1(x) = \tau(x) - \tau_{max} \leq 0, \\
 & g_2(x) = \sigma(x) - \sigma_{max} \leq 0, \\
 & g_3(x) = x_1 - x_4 \leq 0, \\
 & g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0, \\
 & g_5(x) = 0.125 - x_1 \leq 0, \\
 & g_6(x) = \delta(x) - \delta_{max} \leq 0, \\
 & g_7(x) = P - P_c(x) \leq 0,
 \end{aligned} \tag{16}$$

$$\begin{aligned}
 \text{where } \tau(x) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2x_1x_2}}, \quad \tau'' = \frac{MR}{J}, \quad M = \\
 &P(L + \frac{x_2}{2}), \quad R = \sqrt{\frac{x_2^2 + (x_1 + x_3)^2}{4}}, \quad J = 2 \left\{ \sqrt{2x_1x_2} \left[\frac{x_2^2}{12} + \frac{(x_1 + x_3)^2}{4} \right] \right\}, \quad \sigma(x) = \frac{6PL}{x_4x_3^2}, \\
 \delta(x) &= \frac{4PL^3}{E x_3^3 x_4}, \quad P_c(x) = \frac{4.013E \sqrt{x_3^2 x_4^6}}{6L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right).
 \end{aligned}$$

Identical parameters with (Tsipianitis and Tsompanakis 2020) are adopted, with $P = 6000$ 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. The reference objective function value is equal to 1.72485084 lb. The results obtained using the PBA and the various CHTs are given in Table 18, where evidently the best solution was achieved by EconstNEW3, with an objective function value lower of 1.729478, and the lowest CoV by AdPenalty, EconstOR and EconstNEW2, with values below 2%. It should be noted that StochRanking failed to converge to a feasible design solution.

5.2.2 Pressure vessel design

This COP concerns the optimal design of a cylindrical vessel with semispherical heads at its two ends via minimization of the vessel weight, which depends on four design variables: cylindrical vessel thickness x_1 , semispherical head thickness x_2 , and cylindrical vessel inner and outer radii x_3 and x_4 , respectively. The problem can be formulated as follows:

$$\begin{aligned}
 & \text{Minimize } f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \\
 & \text{subject}
 \end{aligned}$$

$$\begin{aligned}
 g_1(x) &= -x_1 + 0.0193x_3 \leq 0, \\
 g_2(x) &= -x_2 + 0.00954x_3 \leq 0, \\
 g_3(x) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\
 g_4(x) &= x_4 - 240 \leq 0.
 \end{aligned}
 \tag{17}$$

Identical parameters with (Tsipianitis and Tsompanakis 2020) are adopted. The reference cost objective function value is equal to 5885.3328 lb (Tsipianitis and Tsompanakis 2020). The results obtained using the PBA and the various CHTs are given in Table 19. *FeasRulesNEW1*, *AdPenalty*, *StochRanking*, *FeasRulesNEW3*, *EconstNEW3* and *EconstNEW4* produced solutions lower than 6000 lb, where the highest performance attained by *AdPenalty* with a value of 5767.50. The lowest CoV value of 3.16% was achieved by *FeasRulesNEW1*.

5.2.3 Tension–compression string design problem

This numerical example concerns the weight minimization of a tension–compression string, which depends on three design variables: string diameter x_1 , mean coil diameter x_2 , and coil number x_3 . The COP is formulated as follows:

$$\begin{aligned}
 & \text{Minimize } f(x) = (x_3 + 2)x_2x_1^2, \\
 & \text{subject} \\
 & g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0, \\
 & g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\
 & g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\
 & g_4(x) = \frac{x_2 + x_1}{1.5} - 1 \leq 0.
 \end{aligned}
 \tag{18}$$

Identical parameters with (Tsipianitis and Tsompanakis 2020) are adopted. The reference objective function value is equal to 0.012665 lb. The results obtained using the PBA and the various CHTs are given in Table 20. Almost all CHTs lead to weight objective function values less than 0.013 lb, with the lowest value of 0.0126911 lb achieved by *FeasRulesNEW1*, while CoV values less than 2.0% were achieved by *FeasRulesNEW4* and *EconstNEW4*.

5.2.4 2D and 3D truss design

Three sizing COPs previously investigated in Farshi and Alinia-ziazi (2010) are examined herein, concerned with the optimal structural design in respect of weight of the 2D 10-bar, 3D 25-bar and 3D 72-bar steel truss structures shown in Figs. 1, 2 and 3.

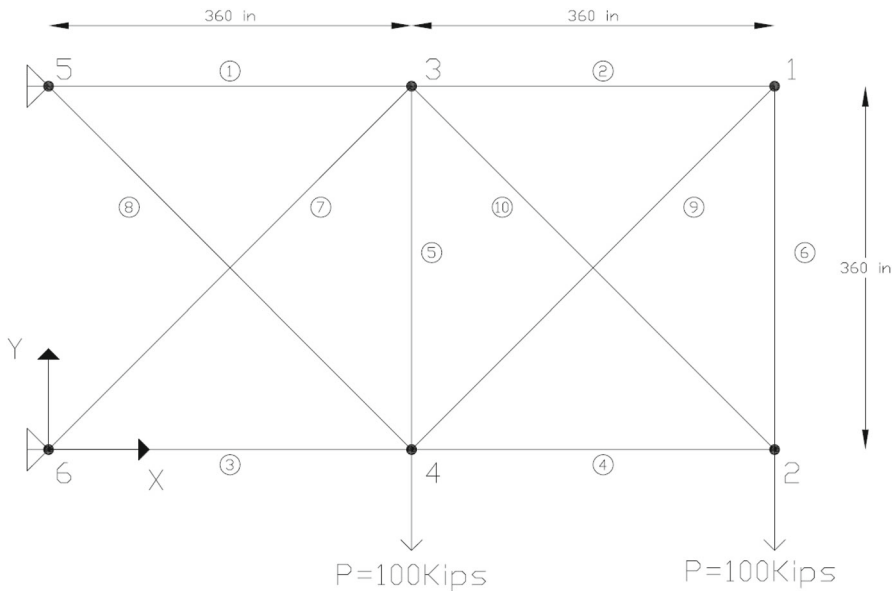


Fig. 1 Geometric, boundary condition, material and loading specifications for 2D 10-bar truss constrained optimization problem

The design variables are the cross-sectional areas of individual structural elements or element groups.

For the 2D 10-bar truss problem shown in Fig. 1 an independent design variable is employed for each bar, continuously defined in the interval $[0.1, 33.5]$ in². The COP is formulated with 28 constraint functions, which impose: (i) tensile and compressive stress limits of 25 ksi for each bar; and (ii) horizontal and vertical displacement limits of 2.0 at each unrestrained node. More details on the problem formulation can be found in Farshi and Alinia-ziazi (2010).

The reference structural weight objective function value found in the literature is equal to 5057.88 lb (Farshi and Alinia-ziazi 2010). The results obtained using the PBA and the various CHTs are given in Table 21. Evidently, most CHTs achieved objective function values less than 5500 lb, with AdPenalty achieving the best solution of 5293.971 lb. *FeasRulesOR*, *FeasRulesNEW1*, *EconstNEW1*, *FeasRulesNEW3* and *FeasRulesNEW4* achieved the lowest CoV, with values below 3.0%. *StochRanking* failed to converge to a feasible design.

For the 3D 25-bar truss problem shown in Fig. 2a–d, the structural elements are allocated to 8 groups, each corresponding to a design variable continuously defined in the interval $[0.01, 3.4]$ in². The CPO is formulated with 110 constraint functions, which impose: (i) a tensile stress limit of 35 ksi and a compressive stress limit in accordance with the AISC code; and (ii) a limit of 0.35 on the absolute value of the displacement components of each unrestrained node. More details on the problem formulation can be found in Farshi and Alinia-ziazi (2010).

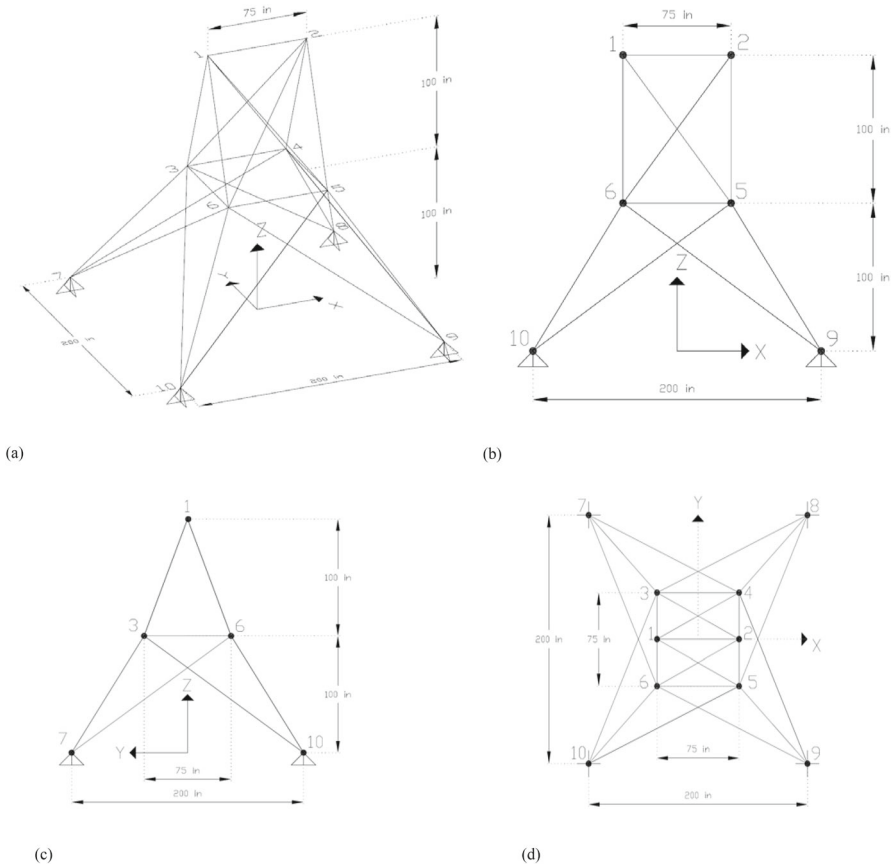


Fig. 2 Geometric, boundary condition, material and loading specifications for 3D 25-bar truss constrained optimization problem: **a** perspective view; **b** elevation on X-Z plane; **c** elevation on Y-Z plane; and **d** plan view on X-Y plane

The reference structural weight objective function value found in the literature is equal to 545.175 lb (Farshi and Alinia-ziazi 2010). The results obtained using the PBA and the various CHTs are given in Table 22. Evidently, most CHTs achieved objective function values less than 560 lb. *FeasRulesNEW2*, *FeasRulesNEW4* and *EconstNEW4* achieved the lowest CoV, with values below 1.0%. *StochRanking* failed to converge to a feasible design.

For the 72-bar truss problem shown in Fig. 3a–d, the structural elements are allocated to 16 groups, each corresponding to a design variable continuously defined in the interval $[0.1, 3.0] \text{ in}^2$. The CPO is formulated with 264 constraint functions which impose: (i) tensile and compressive stress limits of 25 ksi for each bar; and (ii) a limit of 0.35 on the absolute value of the displacement components of each unrestrained node. More details on the problem formulation can be found in Farshi and Alinia-ziazi (2010). The results obtained using the PBA and the various CHTs are given in Table 22.

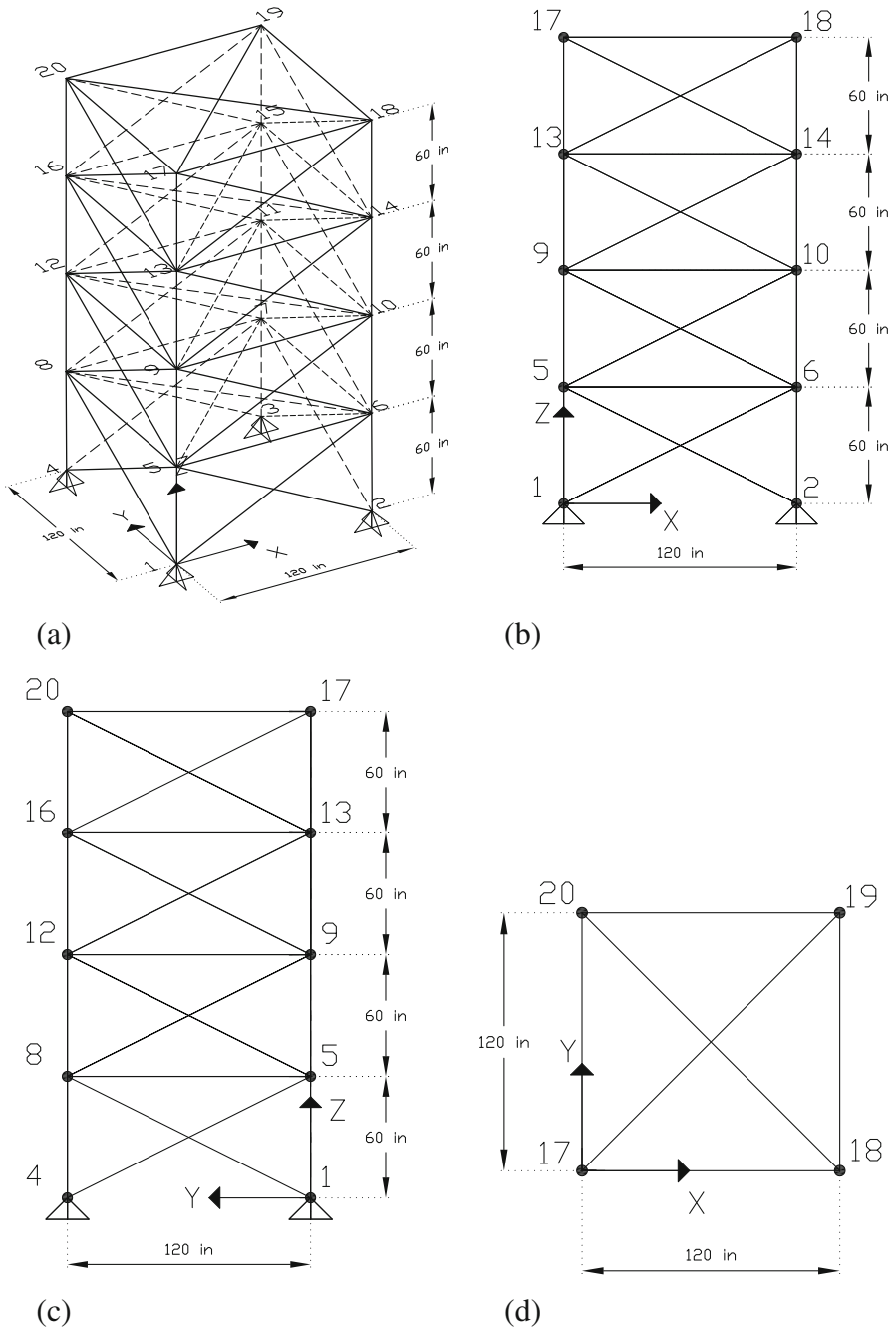


Fig. 3 Geometric, boundary condition, material and loading specifications for 3D 72-bar truss constrained optimization problem: **a** perspective view; **b** elevation on X-Z plane; **c** elevation on Y-Z plane; and **d** plan view on X-Y plane

The reference structural weight objective function value found in the literature is equal to 379.66 lb (Farshi and Alinia-ziazi 2010). The results obtained using the PBA and the various CHTs are given in Table 23. *FeasRulesNEW2* and *EconstNEW2* achieved objective function values less than 400 lb. *AdPenalty* achieved the lowest CoV of 0.54%.

5.3 Discussion

Evidently, the novel CHT variants overall outperform the original ones in most COPs, in respect of both objective function values and CoV. This is attributed to their novel feature of performing individual comparisons based not only upon the objective function value, but also the level of constraint violation and the number of violated constraints. At this point it must be mentioned that the derived conclusions concern the 20 numerical examples examined herein and the specific PBA and CHT variant implementation and can thus not be generalized.

6 Conclusions

This paper presents a comprehensive state-of-the-art review on almost 60 existing CHT variants that can be implemented in MOAs to solve single-objective COPs. Furthermore, eight new variants of the well-known feasibility rules and ε -constrained CHTs are proposed. The distinctive feature of the proposed variants compared to the original ones is that necessary comparisons between individuals depend not only on the objective function value, but also on the level of constraint violation and the number of violated constraint functions. The proposed variants were successfully implemented in the metaheuristic PBA (Kallioras et al. 2018), within HP-OCP, and their performance was evaluated and compared to that of the well-known adaptive penalty (Kawachi et al. 2019), original feasibility rules (Deb 2000), improved ε -constrained (Fan et al. 2018) and original stochastic ranking (Runarsson and Yao 2000) variants, based upon 20 single-objective benchmark mathematical and engineering COPs. The results clearly demonstrate the performance superiority of the proposed novel variants in comparison with the existing ones in most COPs, in respect of both objective function values and CoV values. This highlights the effectiveness and versatility of the proposed approach that considers the level and number of constraint violations for selecting individuals. Concluding, it is important to emphasize that there is no unique constraint handling technique that consistently performs best in any type of problem.

Acknowledgements This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (project code: T1EDK-05603).

Funding Open access funding provided by HEAL-Link Greece.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long

as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ameica-Alducin M-Y, Hasani-Shoreh M, Blaikie W, Neumann F, Mezura-Montes E (2018) A comparison of constraint handling techniques for dynamic constrained optimization problem. In: 2018 IEEE Congress on Evolutionary Computation, CEC 2018—Proceedings, 8477750
- Arouri Y, Sayyafzadeh M (2020) An accelerated gradient algorithm for well control optimization. *J Petrol Sci Eng* 190:106872
- Atamna A, Auger A, Hansen N (2020) On invariance and linear convergence of evolution strategies with augmented Lagrangian constraint handling. *Theor Comput Sci* 832:68–97
- Biedrzycki R (2020) Handling bound constraints in CMA-ES: an experimental study. *Swarm Evol Comput* 52:100627
- Biedrzycki R, Arabas J, Jagodziński D (2019) Bound constraints handling in differential evolution: an experimental study. *Swarm Evol Comput* 50:100453
- Biswas PP, Suganthan PN, Mallipeddi R, Amaratunga GAJ (2018) Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques. *Eng Appl Artif Intell* 68:81–100
- Caraffini F, Kononova AV, Corne D (2019) Infeasibility and structural bias in differential evolution. *Inf Sci* 496:161–179
- Chehouri A, Younes R, Perron J, Ilinca A (2016) A constraint-handling technique for genetic algorithms using a violation factor. *J Comput Sci* 12(7):350–362
- Chootinan P, Chen A (2006) Constraint handling in genetic algorithms using a gradient-based repair method. *Comput Oper Res* 33(8):2263–2281
- Chu S, Yang Z, Xiao M, Qiu H, Gao K, Gao L (2020) Explicit topology optimization of novel polyline-based core sandwich structures using surrogate assisted evolutionary algorithm. *Comput Methods Appl Mech Eng* 369:113215
- Coello CAC, Mezura-Montes E (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inform* 16(3):193–203
- Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inform* 16(3):193–203
- Cucuzza R, Rosso MM, Aloisio A, Melchiorre J, Giudice M, Marano GC (2022) Size and shape optimization of a guyed mast structure under wind, ice and seismic loading. *Appl Sci* 12(10):4875. <https://doi.org/10.3390/app12104875>
- da Silva EK, Barbosa HJC, Lemonge ACC (2011) An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization. *Optim Eng* 12(1–2):31–54
- Datta R, Deb K, Kim J-H (2019) CHIP: Constraint handling with individual penalty approach using a hybrid evolutionary algorithm. *Neural Comput Appl* 31(9):5255–5271
- Deb K (2000) An efficient constraint handling method for genetic algorithms. *Comput Methods Appl Mech Eng* 186(2–4):311–338
- Fan Z, Fang Y, Li W, Yuan Y, Wang Z, Bian X (2018), LSHADE44 with an improved ϵ -constraint-handling method for solving constrained single-objective optimization problems. In: 2018 IEEE congress on evolutionary computation, CEC 2018—proceedings, 8477943
- Farshi B, Alinia-ziazi A (2010) Sizing optimization of truss structures by method of centers and force formulation. *Int J Solids Struct* 47(18–19):2508–2524
- Gandomi AH, Deb K (2020) Implicit constraints handling for efficient search of feasible solutions. *Comput Methods Appl Mech Eng* 363:112917
- Gandomi AH, Kashani AR (2018) Probabilistic evolutionary bound constraint handling for particle swarm optimization. *Oper Res Int J* 18(3):801–823

- Gandomi AH, Yang X-S (2012) Evolutionary boundary constraint handling scheme. *Neural Comput Appl* 21(6):1449–1462
- Javed H, Jan MA, Tairan N, Mashwani WK, Khanum RA, Sulaiman M, Khan HU, Shah H (2019) On the efficacy of ensemble of constraint handling techniques in self-adaptive differential evolution. *Mathematics* 7(7):635
- Jordehi AR (2015) A review on constraint handling strategies in particle swarm optimisation. *Neural Comput Appl* 26(6):1265–1275
- Juárez-Castillo E, Acosta-Mesa H-G, Mezura-Montes E (2019) Adaptive boundary constraint-handling scheme for constrained optimization. *Soft Comput* 23(17):8247–8280
- Kallioras NA, Lagaros ND, Avtzis DN (2018) Pity beetle algorithm – A new metaheuristic inspired by the behavior of bark beetles. *Adv Eng Softw* 121:147–166
- Kaucic M, Barbini F, Camerota Verdù FJ (2020) Polynomial goal programming and particle swarm optimization for enhanced indexation. *Soft Comput* 24(12):8535–8551
- Kawachi T, Kushida J-I, Hara A, Takahama T (2019) Efficient constraint handling based on the adaptive penalty method with balancing the objective function value and the constraint violation. In: 2019 IEEE 11th international workshop on computational intelligence and applications, IWCI 2019—proceedings, 8955094, pp 121–128
- Lagaros ND (2014) A general purpose real-world structural design optimization computing platform. *Struct Multidiscip Optim* 49:1047–1066
- Lagaros ND, Plevris V, Kallioras NA (2022) The mosaic of metaheuristic algorithms in structural optimization. *Arch Comput Methods Eng*. 29: 5457–5492. <https://doi.org/10.1007/s11831-022-09773-0>
- Li X, Xiao C, Lu Z (2018) A new constraint handling method for economic dispatch. *J Electr Eng Technol* 13(3):1099–1109
- Li Z, Chen S, Zhang S, Jiang S, Gu Y, Nouioua M (2019b) FSB-EA: Fuzzy search bias guided constraint handling technique for evolutionary algorithm. *Expert Syst Appl* 119:20–35
- Li S, Gong W, Wang L, Yan X, Hu C (2020) Optimal power flow by means of improved adaptive differential evolution. *Energy* 198:117314
- Li C, Fang H, Gong C (2019a) Expensive inequality constraints handling methods suitable for dynamic surrogate-based optimization. In: 2019a IEEE congress on evolutionary computation, CEC 2019a—proceedings, 8790253, 2010–2017
- Lin X, Luo W, Qiao Y, Xu P, Zhu T (2019) Empirical study of population-based dynamic constrained multimodal optimization algorithms. In: 2019 IEEE symposium series on computational intelligence, SSCI 2019, 9002835, pp 722–730
- Liu S, Zhu H, Chen Z, Cao H (2020) Frequency-constrained truss optimization using the fruit fly optimization algorithm with an adaptive vision search strategy. *Eng Optim* 52(5):777–797
- Malan KM (2018) Landscape-aware constraint handling applied to differential evolution, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11324 LNCS, pp 176–187
- Mallipeddi R, Suganthan PN (2010) Ensemble of constraint handling technique. *IEEE Trans Evol Comput* 14(4):561–579
- Mallipeddi R, Jeyadevi S, Suganthan PN, Baskar S (2012) Efficient constraint handling for optimal reactive power dispatch problems. *Swarm Evol Comput* 5:28–36
- Mao J-Q, Tian M-M, Hu T-F, Ji K, Dai L-Q, Dai H-C (2019) Shuffled complex evolution coupled with stochastic ranking for reservoir scheduling problems. *Water Sci Eng* 12(4):307–318
- Mazhoud I, Hadj-Hamou K, Bigeon J, Joyeux P (2013) Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. *Eng Appl Artif Intell* 26(4):1263–1273
- Melchiorre J, Bertetto A, Marano GC (2021) Application of a machine learning algorithm for the structural optimization of circular arches with different cross-sections. *J Appl Math Phys* 9:1159–1170. <https://doi.org/10.4236/jamp.2021.95079>
- Mezura-Montes E, Coello CAC (2011) Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm Evol Comput* 1(4):173–194
- Miettinen K, Mäkelä MM, Toivanen J (2003) Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms. *J Global Optim* 27(4):427–446
- Miranda-Varela M-E, Mezura-Montes E (2018) Constraint-handling techniques in surrogate-assisted evolutionary optimization. An empirical study. *Appl Soft Comput J* 73:215–229
- Montemurro M, Vincenti A, Vannucci P (2013) The automatic dynamic penalization method (ADP) for handling constraints with genetic algorithms. *Comput Methods Appl Mech Eng* 256:70–87

- Mozaffari A, Emami M, Fathi A (2019) A comprehensive investigation into the performance, robustness, scalability and convergence of chaos-enhanced evolutionary algorithms with boundary constraints. *Artif Intell Rev* 52(4):2319–2380
- Peng C, Liu H-L, Gu F (2018) A novel constraint-handling technique based on dynamic weights for constrained optimization problems. *Soft Comput* 22(12):3919–3935
- Qian J, Yi J, Cheng Y, Liu J, Zhou Q (2020) A sequential constraints updating approach for Kriging surrogate model-assisted engineering optimization design problem. *Eng Comput* 36(3):993–1009
- Rodrigues MDC, Guimarães S, de Lima BSLP (2018) E-BRM: A constraint handling technique to solve optimization problems with evolutionary algorithms. *Appl Soft Comput J* 72:14–29
- Rodrigues LL, Sebastian Solis-Chaves J, Vilcanqui OAC, Filho AJS (2020) Predictive incremental vector control for DFIG with weighted-dynamic objective constraint-handling method-PSO weighting matrices design. *IEEE Access* 8:114112–114122
- Rosso MM, Cucuzza R, Aloisio A, Marano GC (2022) Enhanced multi-strategy particle swarm optimization for constrained problems with an evolutionary-strategies-based unfeasible local search operator. *Appl Sci* 12(5):2285. <https://doi.org/10.3390/app12052285>
- Rosso MM, Cucuzza R, Di Trapani F, Marano GC (2021) Nonpenalty machine learning constraint handling using PSO-SVM for structural optimization. *Adv Civ Eng*, Article ID 6617750, <https://doi.org/10.1155/2021/6617750>
- Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans Evol Comput* 4(3):284–294
- Salcedo-Sanz S (2009) A survey of repair methods used as constraint handling techniques in evolutionary algorithms. *Comput Sci Rev* 3(3):175–192
- Shabani A, Asgarian B, Salido M, Asil Gharebaghi S (2020) Search and rescue optimization algorithm: a new optimization method for solving constrained engineering optimization problems. *Expert Syst Appl* 161:113698
- Stanovov V, Akhmedova S, Semenkin E (2020) Combined fitness–violation epsilon constraint handling for differential evolution. *Soft Comput* 24(10):7063–7079
- Takahama, T, Sakai, S (2005) Constrained optimization by ϵ constrained particle swarm optimizer with ϵ -level control. In: *Advances in Soft Computing (AISC)*, pp 1019–1029
- Takahama T, Sakai S (2010) Constrained optimization by the ϵ -constrained differential evolution with an archive and gradient-based mutation. In: *2010 IEEE World Congress on Computational Intelligence, WCCI 2010–2010 IEEE Congress on Evolutionary Computation, CEC 2010*, 5586484
- Trivedi A, Biswas N, Chakroborty S, Srinivasan D (2018) Extending unified differential evolution with a new ensemble of constraint handling techniques. In: *2017 IEEE symposium series on computational intelligence, SSCI 2017—proceedings*, pp 1–8
- Tsipianitis A, Tsompanakis Y (2020) Improved cuckoo search algorithmic variants for constrained nonlinear optimization. *Adv Eng Softw* 149:102865
- Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct Multidiscip Optim* 37(4):395–413
- Wang S, Kang J, Tasgetiren MF, Gao L, Kizilay D (2019) Differential evolution through variable neighborhood search for constrained real parameter optimization problems. In *2019 IEEE symposium series on computational intelligence, SSCI 2019*, 9003026, pp 2252–2260
- Zein S, Madhavan V, Dumas D (2017) A repair operator for the preliminary design of a composite structure using genetic algorithms. *Struct Multidiscip Optim* 55(6):2099–2110
- Zhang C, Lin Q, Gao L, Li X (2015) Backtracking Search Algorithm with three constraint handling methods for constrained optimization problems. *Expert Syst Appl* 42(21):7831–7845
- Zhao X, Zhang K, Chen G, Xue X, Yao C, Wang J, Yang Y, Zhao H, Yao J (2020) Surrogate-assisted differential evolution for production optimization with nonlinear state constraints. *J Petrol Sci Eng* 194:107441