

Constrained Optimization with Evolutionary Algorithms: A Comprehensive Review

Ali Osman Kusakci ^a, Mehmet Can ^b

^a Faculty of Engineering and Natural Sciences, International University of Sarajevo, Hrasnicka Cesta 15, 71210, Ilidza, akusakci@ius.edu.ba

^b Faculty of Engineering and Natural Sciences, International University of Sarajevo, Hrasnicka Cesta 15, 71210, Ilidza, mcan@ius.edu.ba

Abstract— (Global optimization is an essential part of any kind of system. Various algorithms have been proposed that try to imitate the learning and problem solving abilities of the nature up to certain level. The main idea of all nature-inspired algorithms is to generate an interconnected network of individuals, a population. Although most of unconstrained optimization problems can be easily handled with Evolutionary Algorithms (EA), constrained optimization problems (COPs) are very complex. In this paper, a comprehensive literature review will be presented which summarizes the constraint handling techniques for COPs.)

Keywords—constrained optimization, evolutionary algorithms, nonlinear programming, genetic algorithms

INTRODUCTION

Global optimization is an essential part of any engineering, economic and social system. Since Holland's (Holland, 1975) ground breaking work, global optimization approaches inspired by nature have been widely used in global optimization. Various algorithms have been proposed that try to imitate the learning and problem solving abilities of the nature up to certain level. Ant colony optimization (ACO), particle swarm optimization (PSO) (J J Liang, Zhigang, & Zhihui, 2010), artificial immune systems (Farmer, Packard, & Perelson, 1986), evolutionary algorithms (EA) (Ho & Shimizu, 2007), artificial bee colony (ABC) (Karaboga & Akay, 2011), estimation of distribution algorithms (EDA) (Bi & Zhang, 2011; Larrañaga & Lozano, 2002) are just a few of them to be mentioned.

The main idea of all nature-inspired algorithms is to generate an interconnected network of individuals, a population. It is assumed that the interactions between parallel working agents can be exploited, as this will generate more than each individual can bring forward alone. Namely, they represent the 1+1>2 effect. This is in general called *collective intelligence*. The population based algorithms are known for their global search ability and very precise approximation of global solutions despite their relatively slow convergence for some problems and approximation bias (Yang, Xu, & Soh, 2006).

Although most of unconstrained optimization problems with moderate to high dimensions can be easily handled with Evolutionary Algorithms (EA), constrained optimization problems (COPs) with inequality and equality constraints are very hard to deal with (Hamida & Petrowski, 2000). The difficulty level also depends on the dimension, number of inequality and equality constraints as well as structural specifications of the problem, including sparsity of the feasible domain, position of the global solution (for instance: a solution lying on the boundary of feasible domain), non-separable character of the variables and nonlinear structure of the objective function. Thus, COPs require exhaustive search of the feasible domain (Hamida & Petrowski, 2000; Michalewicz & Schoenauer, 1996).

Despite the fact that there have been numerous constraint handling techniques proposed by researchers (Mezura-Montes & Coello Coello, 2011), there is still a need to design new methods, which have to be computationally efficient and reliable (Coello Coello & Mezura Montes, 2002). In the design of new algorithms most researchers have been focused to determine how to generate feasible individuals while maintaining a reasonable ratio between feasible and infeasible members in a population so that the algorithm is able to jump in a sparse feasible domain (Hamida & Schoenauer, 2002; Ho & Shimizu, 2007; Runarsson, 2000). Namely, all algorithms

aim to find a balance between exploration-, exploitation-power, and feasibility.

CONSTRAINED OPTIMIZATION

2.1. Basic Concepts

An n dimensional COP can be defined by two components: an objective function to be maximized or minimized, and several inequality and equality constraints (Mezura-Montes & Coello Coello, 2011). The general structure is defined as:

$$\begin{aligned} \min \text{ or } \max \quad & f(\vec{x}), \quad \vec{x} = [x_1, \dots, x_n]^T \\ & \in F \subseteq S \subseteq \mathbb{R}^n \\ \text{subject to} \quad & \end{aligned}$$

$$\begin{aligned} g_i(\vec{x}) &\leq 0, \quad i = 1, \dots, r \\ h_i(\vec{x}) &= 0, \quad i = r + 1, \dots, m, \end{aligned}$$

Where $S = \{\vec{x} \in \mathbb{R}^n | l_j \leq x_j \leq u_j\}$, $j = 1, \dots, n$ and $F = \{\vec{x} \in S | g_i(\vec{x}) \leq 0 \text{ and } h_i(\vec{x}) = 0\}$, \vec{x} is solution vector $\vec{x} = [x_1, \dots, x_n]^T$, r is the number of inequality and $m-r$ is the number of equality constraints. Some researchers convert equality constraints into inequalities by adding a small tolerance $\epsilon > 0$.

$$|h_i(\vec{x})| - \epsilon \leq 0$$

The sum of constraint violations can be formulated as follows:

$$v(\vec{x}) = \sum_{i=1}^r \max(0, g_i(\vec{x})) + \sum_{i=r+1}^m |h_i(\vec{x})|$$

The measure of quality of a solution \vec{x} can be defined with three terms: (1) objective function value $f(\vec{x})$, (2) sum of constraint violations $v(\vec{x})$, and (3) number of constraints violated $\rho(\vec{x})$. This three factors or their various combinations are usually employed as quality metric in constrained optimization literature (Ho & Shimizu, 2007; Mezura-Montes & Coello Coello, 2011).

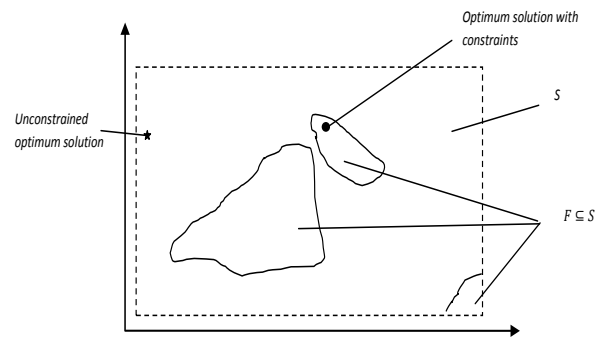


Figure 1: Illustrative example of a COP in two dimensions

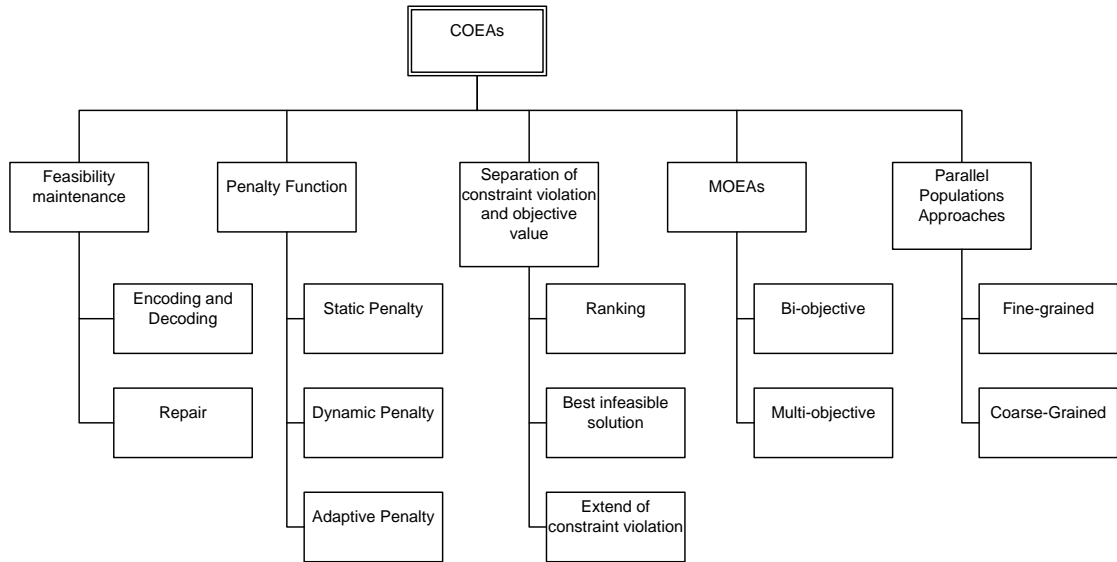


Figure 2: The taxonomy of COEAs

2.1. Basic Concepts

As mentioned before, various different techniques have been proposed to handle COPs. An extended survey can be found in (Mezura-Montes & Coello Coello, 2011) and (Yu & Gen, 2010). The constrained optimization evolutionary algorithms (COEAs) can be classified in the following five categories illustrated in Figure 2: feasibility maintenance, penalty function, separation of constraint violation and objective function, multi-objective optimization evolutionary algorithms (MOEA), parallel populations based methods.

Feasibility Maintenance

Approaches based on feasibility maintenance aim to bring the individuals to the feasible domain. Repairing infeasible individuals and homomorphous mapping are two methods that dominate this category. The repaired individuals are replaced or sometimes used only for evaluation purposes (Koziel & Michalewicz, 1996). To repair infeasible individuals, problem specific operators must be designed, which may not be an efficient method in some cases and repair operator may introduce a strong bias in the search. This may harm the evolutionary process itself (C. a. Coello Coello & Mezura Montes, 2002). Homomorphous mapping tries to maintain the feasibility of population by mapping the feasible domain onto a hypercube and performing evolutionary operators within the hypercube. The offspring, guaranteed to be feasible, are then transferred back to the definition domain (Koziel & Michalewicz, 1996). Despite its secure feasibility maintenance property, homomorphous mapping comes along with high computational cost because of back and forward mapping must be conducted through some optimization methods for each individual (Yu & Gen, 2010), (Koziel & Michalewicz, 1996).

Penalty Function

The methods based on penalty functions are the most popular approaches, thanks to their simplicity and easy application (Hamida & Schoenauer, 2002). They rely on

penalizing the individuals, which are out of the feasible domain, so that a feasible point will be superior to an infeasible point of comparable fitness. However, two main questions arise in penalty based method:

- How to adjust the penalty weights related to the constraints.
- How to maintain a certain percentage of infeasible individuals in the population, which allows determining the global optimum in highly sparse feasible space.

The penalty weights must be tuned very carefully in order to avoid the above mentioned two problems. A small penalty level leads to solutions, which are infeasible (some penalized points may still have better penalized fitness than the best feasible point); on the other hand, high penalty levels restrict the search inside the feasible region, forbidding any short-cut across the infeasible region, and thus eventually failing to converge to the optimal solution. Various methods have been developed to overcome the above-mentioned burdens. Penalty based approaches can be classified as static (we consider death penalty approaches under static penalty methods) (Hoffmeister & Sprave, 1996), (Homaifar, Qi, & Lai, 1994), dynamic (C. Coello Coello, 2002), and adaptive (Hamida & Schoenauer, 2002), (Hadj-Alouane & Bean, 1997), (Hinterding, 2001) penalty function methods. Penalty function method introduced in (Smith & Tate, 1993) is improved and embedded by Tasgetiren and Suganthan (2006) into a multi-populated DE where they introduced near feasibility threshold (NFT) mechanism in which the NFT region is considered a promising search region beyond the feasible region.

Separation of Objective Function and Constraints

The third class in constrained optimization separates the objective value and the constraint violation. In the replacement procedure, Deb (2000) suggested three comparison rules between two individuals to maintain the balance between feasible and infeasible individuals. Thus, feasible and infeasible individuals are evaluated with different criteria. A

binary tournament selection method is embedded to a GA with following three comparison rules:

1. If both solutions are feasible, the one with the best objective function must be chosen.
2. If one solution is feasible while the other is infeasible, the feasible one must be chosen.
3. If both solutions are infeasible, the one with the lowest sum of constraint violation must be chosen.

These rules are referred in the literature as feasibility rules and adopted with many EAs including DE (Brest, Zumer, & Maucec, 2006), real coded GA (Sinha, Srinivasan, & Deb, 2006). Munoz-Zavala et al. (2006) used the feasibility rules with Particle Evolutionary Swarm Optimization Plus (PESO+) and introduced an archive keeping the so called tolerance particles which were able to survive after the reduction of the tolerance used for equality constraints throughout some generations in the swarm. ABC algorithm is modified for constrained optimization problems by Karaboga and Akay (2011) for constraint handling, and as constraint handling mechanism the feasibility rules are used. The algorithm is tested on thirteen well-known test problems and the results obtained are compared to those of the state-of-the-art algorithms.

Another application of feasibility rules is done by (Zielinski & Laur, 2006a) and (Zielinski & Laur, 2006b) where two versions, a DE and a PSO implementation respectively, are tested on 24 benchmark problems. The idea of separating of objective function and constraints has led to the approach of assigning each constraint and objective function to a sub-population with continuous information exchange between them (J.J. Liang & Suganthan, 2006).

Another pioneering alternative inspired by the separation of objective function and constraints was developed by Runarsson (2000, 2006) and named as Stochastic Ranking (SR). They introduced a probability value P_f to compare infeasible individuals based on their fitness values. That is, given any pair of two adjacent individuals, they are compared according to their objective function values with probability 1 in case both are feasible; otherwise, this probability is P_f . SR has been further developed in (Runarsson, 2006) where Runarsson focused on using fitness approximation, surrogate modeling, for constrained numerical optimization. In this approach k-nearest-neighbors (NN) regression is coupled to SR. SR was originally designed with an ES search algorithm and has been applied to different domains (Mezura-Montes & Coello Coello, 2011). SR is proved to be efficient and highly competitive with other methods (Yu & Gen, 2010). As a general concept of constraint handling SR have been accompanied with other evolutionary algorithms including DE (Mezura-Montes, Velázquez-Reyes, & Coello Coello, 2005) and ACO (Leguizamón & Coello Coello, 2007).

One of the most recent constraint-handling techniques stated in the literature is the ϵ -constrained method proposed by Takahama, Sakai (2005). This method converts a COP into an unconstrained optimization problem. It has two main components: (1) a relaxation of the limit to consider a solution as feasible, based on its sum of constraint violation, with the

aim of using its objective function value as a comparison criterion, and (2) a lexicographical ordering mechanism in which the minimization of the sum of constraint violation precedes the minimization of the objective function of a given problem. The value of $\epsilon > 0$ determines the so-called ϵ -level comparisons between a pair of solutions \vec{x}_1 and \vec{x}_2 with objective function values $f(\vec{x}_1)$ and $f(\vec{x}_2)$ and sum of constraint violations $v(\vec{x}_1)$ and $v(\vec{x}_2)$ as given in equations (1) and (2).

$$\begin{aligned} & (f(\vec{x}_1), v(\vec{x}_1)) <_{\epsilon} (f(\vec{x}_2), v(\vec{x}_2)) \\ \Leftrightarrow & \begin{cases} f(\vec{x}_1) < f(\vec{x}_2), & \text{if } v(\vec{x}_1), v(\vec{x}_2) \leq \epsilon \\ f(\vec{x}_1) < f(\vec{x}_2), & \text{if } v(\vec{x}_1) = v(\vec{x}_2) \\ v(\vec{x}_1) < v(\vec{x}_2), & \text{otherwise} \end{cases} \end{aligned} \quad (1)$$

$$\begin{aligned} & (f(\vec{x}_1), v(\vec{x}_1)) \leq_{\epsilon} (f(\vec{x}_2), v(\vec{x}_2)) \\ \Leftrightarrow & \begin{cases} f(\vec{x}_1) \leq f(\vec{x}_2), & \text{if } v(\vec{x}_1), v(\vec{x}_2) \leq \epsilon \\ f(\vec{x}_1) \leq f(\vec{x}_2), & \text{if } v(\vec{x}_1) = v(\vec{x}_2) \\ v(\vec{x}_1) < v(\vec{x}_2), & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

The ϵ -constrained method is employed in a competition on constrained real-parameter optimization in 2006 (Takahama & Sakai, 2006), where a DE variant and a gradient-based mutation operator were employed. An archive is used to store feasible elites in the population. This algorithm was able to obtain the best overall results in the competition, in which a set of 24 test problems were solved (J J Liang et al., 2006). Evidently, the main drawback of this approach is that gradient information must be computed.

Further improvements have been proposed to the ϵ -constrained method in Sakai & Takahama (2007) and (2010). They proposed an adaptive approach for the ϵ value, which allowed a faster decrease in its value. A further improvement of the aforementioned algorithm is proposed by Takahama and Sakai in (Takahama & Sakai, 2010), where an elitist strategy is followed and an archive to store solutions is created. The approach has provided one of the most competitive performances in a newly organized competition on constrained real-parameter optimization with 18 scalable problems in 2010.

Multi-objective concepts

All multi-objective concepts in constrained optimization with single objective function rely on the idea of transforming the given constraints into additional objective functions to be minimized. Thus, the original COP is converted to unconstrained multi-objective optimization problem. Although the violation of each constraint can be handled as a separate objective the common approach is to use sum of constraint violations as a second objective along with the original objective function (Mezura-Montes & Coello, 2008) (Isaacs, Ray, & Smith, 2008). Hence, the problem becomes a bi-objective optimization problem with the aim of finding the Pareto-frontier (Yu & Gen, 2010) as illustrated in Figure 3.

Mezura-Montes et al., (2005) proposed a MOEA based on niched-pareto genetic algorithm (NPGA) utilizing dominance based tournament selection. The algorithm does not require the

use of penalty function and uses niching methods to maintain diversity in the population. They incorporated a selection ratio which indicates the minimum number of individuals that will not be selected with tournament selection and applied the method to benchmark problems in mechanical design domain.

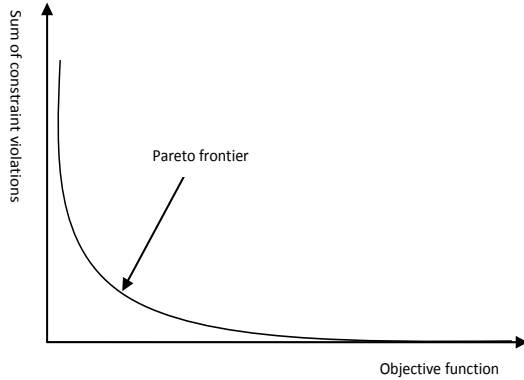


Figure 3: A simple multi-objective model for COPs

Motivated by the idea of keeping good infeasible solutions, Ray et al. (2009) designed the Infeasibility Driven Evolutionary Algorithm (IDEA) whose replacement process requires the definition of a proportion of infeasible solutions to remain in the population for the next generation. This method converts a COP with m constraints into an unconstrained multi-objective optimization problem with *two* objectives. Beside the original objective function, the “violation measure” of the solutions is incorporated as an additional objective. Each solution is ranked for each constraint, and the violation measure is computed as the sum of ranks per solution. A user-defined parameter α is used to maintain a set of infeasible solutions as a fraction of the size of the population. IDEA is applied by Singh et al. (2009) to dynamic single objective optimization problems due to its fast convergence properties reported in (Ray et al., 2009). The algorithm gave promising results on two dynamic single objective problems. However, the method is tested on a very limited number of problems and needs more comprehensive studies justifying its superiority.

Infeasibility Empowered Memetic Algorithm (IEMA) (Singh, Ray, & Smith, 2010) is proposed based on IDEA algorithm with an additional local search operator, sequential quadratic programming (SQP), to increase the local search ability of IDEA. IEMA was tested in eighteen test problems in 10D and 30D. However, the local search algorithm employed needs explicit calculation of gradient.

Li et al. (2008) designed a PSO algorithm in which Pareto dominance was used as a criterion in the velocity update process. The algorithm employed also a small tolerance to deal with slightly feasible solutions and handle them as feasible. Three engineering design problems are solved by Li et al. (2008) and very competitive results are obtained.

Parallel Populations Approaches

a. Fine-Grained Parallelism

Although parallel computation is a mature approach, the idea of parallel working constraint-handling techniques is relatively new. While analyzing this idea the extent of communication in the parallel structure must be studied in details. The communication may be performed in a continuous way where an ensemble of methods is built to obtain better performance.

One of the algorithms implemented the fine-grained parallelism idea successfully is the so called ensemble of constraint handling techniques (ECHT). Motivated by the no free lunch theorem (Wolpert & Macready, 1997), Mallipeddi and Suganthan (2010a) stated that it is impossible for a single constraint handling technique to outperform all other techniques on every problem. ECHT combines four constraint-handling techniques: feasibility rules, stochastic ranking, a self-adaptive penalty function and the ϵ -constrained method. The algorithm employed four subpopulations with close communication. Sub-populations share all of their offspring and all offspring are evaluated with all four different constraint handling methods. The algorithm also applied to the benchmark problems for CEC2010 (R Mallipeddi & Suganthan, 2010b). The results obtained with ECHT were highly competitive (Rammohan Mallipeddi & Suganthan, 2010) (Tasgetiren, Suganthan, Pan, Mallipeddi, & Sarman, 2010). However, the main drawback of the approach is the calibration required for each of the constraint-handling techniques adopted.

Elsayed et al. (2011) proposed a similar DE-based algorithm. The algorithm combines four DE-mutations, two DE recombination and two constraint-handling techniques. Thus, sixteen different combinations of strategies are used to generate offspring in a single-population algorithm. The strategies are called as “donors”. The initial stage assumes that each donor has the same selection probability whereas the improvement made in the population by each donor is recorded and used to determine promising combinations. By this way, better donors are allowed to generate more offspring than low performance donors. The approach was tested on a set of 18 test problems with 10 and 30 dimensions, showing a very competitive performance. Its main drawback is the number of parameters to be tuned by the user.

b. Coarse-Grained Parallelism

Island Models (IM) are classified as coarse-grained parallel algorithms as the communication between subpopulations is highly limited. Coarse-grained parallel algorithms are originally designed to work multi-processor systems to exploit their high computation powers as a single processor was insufficient to deal with optimization problems in large scale (Z. Skolicki, 2005). However, they introduce some basic differences of behavior, which improve their performance even when they are executed on a single processor (J J Liang et al., 2010). Thus, besides reducing execution time by taking advantage of the computational power of parallel machines, they achieve higher quality solutions. Very frequently, the use of a structured population in the form of islands or demes is

responsible for such benefits. The island models are in general considered under parallel evolutionary algorithms (PEAs) (de la Ossa, Gámez, & Puerta, 2006). Their superior performance compared to ordinary EAs can be explained in terms of improved balance between exploitation and exploration of the solution space. In the Island Model, each island can exchange information with its neighbor island as defined in the possible inter-island links commonly referred to as migration topology (Ruciński, Izzo, & Biscani, 2010).

A DE based island model, differential evolution with separated groups (DE-SG), for unconstrained multi-dimensional domain is proposed in (Piotrowski, Napiorkowski, & Kiczko, 2012). The model aims to improve performance for difficult problems by distributing individuals into small sub-populations and defining diversified communication rules between them. The main goal is to make the optimization technique less vulnerable to entrapment in a local minimum and to improve the ability of the algorithm to adapt to various kinds of problems, including rotated ones (Piotrowski et al., 2012). The population of individuals in DE-SG is divided into halves (rules of migration of individuals are different in each half) and then each half is further divided into small groups (or sub-populations) that operate independently. DE-SG works with a special exploration tool. If a particular group has no improvement, the group's members are allowed to communicate with the whole population for some time; doing so allows them to get the necessary knowledge needed to escape from local optimum. The algorithm maintains also one elite group which is allowed to communicate continuously with all other sub-population. Thus, the elites have access to all collected information so far (Piotrowski et al., 2012). The findings of (Piotrowski et al., 2012) indicates that DE-SG outperforms the other methods for vast majority of problems which are rarely or never solved while CMA-ES (Hansen & Ostermeier, 2001) and AMALGAM (Vrugt, Robinson, & Hyman, 2009) are other two effective methods especially for low dimensional problems. However, this method is only tested on unconstrained optimization problems.

Pereira and Lapa (2003) designed so called Island Genetic Algorithm (IGA) and used to optimize Nuclear Power Plant Auxiliary Feedwater System surveillance tests policy. They observe that diversity maintenance in the search process can be improved by means of island models. Furthermore, the obtained results confirmed the superiority of Island Models to the conventional methods.

The issue of controlling values of various parameters of an evolutionary algorithm is one of the most important research areas in evolutionary computation. Srinivasa et al. (2007) addressed the adaptive strategies for parameter control in Island Models and proposed adaptive migration (island) model of GA. The algorithm is built such that, given a search space, the number of individuals in the population that resides in a relatively high fitness region of the search space increases thus improving exploitation. For this high fitness population, the mutation rate and number of points of crossover are decreased to increase exploitation. On the other hand, for populations in relatively low fitness regions, the number of individuals is

decreased but the mutation rates and number of crossover points is increased to make the search of these regions more explorative. Montero and Riff (2011) proposed two new parameter control strategies for evolutionary algorithms based on the ideas of reinforcement learning. They obtained very efficient and low-cost adaptive techniques for parameter control. Another prominent characteristic of the techniques is preservation of the original design of the evolutionary algorithm.

Current literature surveys (Mezura-Montes & Coello Coello, 2011), (C. Coello Coello, 2002), do not indicate any applications of Island Models in constrained optimization domain. Skolick (2007) indicated in his PhD thesis that issue of constraints should be addressed by the island models. In a COP, a sparse feasible area of the search space can be described as a sum of much more regular smaller subareas, which in turn may correspond to particular islands. Thus, each island can explore a feasible region in highly sparse feasible domains. If this was a case, topology of connections between islands would be similar to the general view of genotypic search space. Furthermore, he added that it is not obvious what extensions to island model would be useful for COPs.

CONCLUSION

The literature review presented here summarizes the constraint handling techniques for COPs. The most representative state-of-the-art constraint-handling methods were briefly introduced and discussed. Based on the survey in this section, the following conclusions can be drawn:

A wide variety of approaches was developed with different evolutionary computation tools.

The mostly adopted constraint handling methods can be listed as feasibility rules, and stochastic ranking due their relative simple and effective combination possibilities with various EAs.

The most popular algorithms used in the early years are GAs and ESs while DE has become more popular in the recent years in COP domain. Another search engine employed very frequently is PSO with several variations. The distinct characteristic of both methods is their solution generation scheme, where the solutions are not discarded after each generation but rather changed by the virtue of another variable (Ma, 2010).

Penalty functions are still popular due to their easy incorporation to any EA. The adaptive penalty methods are mostly preferred as they offer a flexible framework for a wide range of COPs and easily implemented.

Another direction in constrained optimization is the hybridizing two or more methods to achieve better performance by combining the strengths of each method. Memetic algorithms, where local search methods are patched to global search algorithms, are another hot topic.

Although fine-grained parallel EAs have been successfully applied to this domain in recent studies (ensemble of constraint handling methods), the scientific work in coarse-grained parallel EAs (island models) for constrained optimization is almost inexistent.

REFERENCES

- Bi, L., & Zhang, S. (2011). Analysis and research models of the Estimation of Distribution Algorithms. *Proceedings of 2011 International Conference on Computer Science and Network Technology* (pp. 2014-2018). IEEE. doi:10.1109/ICCSNT.2011.6182366
- Brest, J., Zumer, V., & Maucec, M. S. (2006). Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. *2006 IEEE International Conference on Evolutionary Computation* (pp. 215-222). Ieee. doi:10.1109/CEC.2006.1688311
- Coello Coello, C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12), 1245-1287. doi:10.1016/S0045-7825(01)00323-1
- Coello Coello, C. a., & Mezura Montes, E. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3), 193-203. doi:10.1016/S1474-0346(02)00011-3
- Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2011). Integrated strategies differential evolution algorithm with a local search for constrained optimization. *2011 IEEE Congress of Evolutionary Computation (CEC)* (pp. 2618-2625). IEEE. doi:10.1109/CEC.2011.5949945
- Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1-3), 187-204. doi:10.1016/0167-2789(86)90240-X
- Hadj-Alouane, A. B., & Bean, J. C. (1997). A Genetic Algorithm for the Multiple-Choice Integer Program. *Operations Research*, 45(1), 92-101.
- Hamida, S. B., & Petrowski, A. (2000). The need for improving the exploration operators for constrained optimization problems. *IEEE Evolutionary Computation* (pp. 1176 - 83). La Jolla, CA.
- Hamida, S. B., & Schoenauer, M. (2002). ASCHEA: New Results Using Adaptive Segregational Constraint Handling. *2002 Congress On Evolutionary Computation IEEE* (pp. 884-889).
- Hansen, N., & Ostermeier, a. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2), 159-95. doi:10.1162/106365601750190398
- Hinterding, R. (2001). Constrained parameter optimisation: equality constraints. *IEEE congress on evolutionary computation* (pp. 687 - 692). Seoul , South Korea: IEEE.
- Ho, P. Y., & Shimizu, K. (2007). Evolutionary constrained optimization using an addition of ranking method and a percentage-based tolerance value adjustment scheme. *Information Sciences*, 177(14), 2985-3004. doi:10.1016/j.ins.2007.01.011
- Hoffmeister, F., & Sprave, J. (1996). Problem-independent handling of constraints by use of metric penalty functions. *Fifth Annual Conference on Evolutionary Programming (EP'96)*. MIT Press.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbo, MIr: The University of Michigan Press.
- Homaifar, A., Qi, C. X., & Lai, S. H. (1994). Constrained Optimization Via Genetic Algorithms. *SIMULATION*, 62(4), 242-253. doi:10.1177/003754979406200405
- Isaacs, A., Ray, T., & Smith, W. (2008). Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)* (pp. 2780-2787). IEEE. doi:10.1109/CEC.2008.4631171
- Karaboga, D., & Akay, B. (2011). A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing*, 11(3), 3021-3031. doi:10.1016/j.asoc.2010.12.001
- Koziel, S., & Michalewicz, Z. (1996). Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary computation*. doi:abs/10.1162/evco.1996.4.1.1
- Larrañaga, P., & Lozano, J. A. (2002). *No Estimation of Distribution Algorithms Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston: Kluwer Academic Publishers.
- Leguizamón, G., & Coello Coello, C. A. (2007). A boundary search based ACO algorithm coupled with stochastic ranking. *2007 IEEE Congress on Evolutionary Computation* (pp. 165-172). IEEE. doi:10.1109/CEC.2007.4424468
- Li, L. D. (2008). A multi-objective constraint-handling method with PSO algorithm for constrained engineering optimization problems. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)* (pp. 1528-1535). IEEE. doi:10.1109/CEC.2008.4630995
- Liang, J J, Runarsson, T. P., Clerc, M., Suganthan, P. N., Coello, C. A. C., & Deb, K. (2006). *Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization*. *Evolutionary Computation* (pp. 251-256).
- Liang, J J, Zhigang, S., & Zhihui, L. (2010). Coevolutionary Comprehensive Learning Particle Swarm Optimizer. *IEEE Congress on Evolutionary Computation* (Vol. 450001, pp. 1-8). IEEE. doi:10.1109/CEC.2010.5585973
- Liang, J.J., & Suganthan, P. N. (2006). Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint-Handling Mechanism. *2006 IEEE International Conference on Evolutionary Computation* (Vol. 639798, pp. 9-16). IEEE. doi:10.1109/CEC.2006.1688284

- Ma, H. (2010). An analysis of the equilibrium of migration models for biogeography-based optimization. *Information Sciences*, 180(18), 3444-3464. doi:10.1016/j.ins.2010.05.035
- Mallipeddi, R., & Suganthan, P. N. (2010a). Ensemble of Constraint Handling Techniques. *IEEE Transactions on Evolutionary Computation*, 14(4), 561-579. doi:10.1109/TEVC.2009.2033582
- Mallipeddi, R., & Suganthan, P. N. (2010b). *Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization*. Singapore.
- Mallipeddi, Rammohan, & Suganthan, P. N. (2010). Differential evolution with ensemble of constraint handling techniques for solving CEC 2010 benchmark problems. *IEEE Congress on Evolutionary Computation* (pp. 1-8). IEEE. doi:10.1109/CEC.2010.5586330
- Mezura-Montes, E., & Coello, C. A. C. (2008). Constrained optimization via multiobjective evolutionary algorithms. In D. C. Joshua Knowles & K. Deb (Eds.), *Multiobjective Problems Solving from Nature: From Concepts to Applications* (pp. 53-76). Springer-Verlag.
- Mezura-Montes, E., & Coello Coello, C. A. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 1(4), 173-194. doi:10.1016/j.swevo.2011.10.001
- Mezura-Montes, E., Velázquez-Reyes, J., & Coello Coello, C. A. (2005). Promising infeasibility and multiple offspring incorporated to differential evolution for constrained optimization. *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05* (p. 225). New York, New York, USA: ACM Press. doi:10.1145/1068009.1068043
- Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4, 1-32.
- Montero, E., & Riff, M.-C. (2011). On-the-fly calibrating strategies for evolutionary algorithms. *Information Sciences*, 181(3), 552-566. doi:10.1016/j.ins.2010.09.016
- Pereira, C. M. N. a., & Lapa, C. M. F. (2003). Parallel island genetic algorithm applied to a nuclear power plant auxiliary feedwater system surveillance tests policy optimization. *Annals of Nuclear Energy*, 30(16), 1665-1675. doi:10.1016/S0306-4549(03)00131-2
- Piotrowski, A. P., Napiorkowski, J. J., & Kiczko, A. (2012). Differential Evolution algorithm with Separated Groups for multi-dimensional optimization problems. *European Journal of Operational Research*, 216(1), 33-46. doi:10.1016/j.ejor.2011.07.038
- Ray, T., Singh, H. K., Isaacs, A., & Smith, W. (2009). Infeasibility driven evolutionary algorithm for constrained optimization. *Constraint-Handling in Evolutionary Optimization* (pp. 145-165). Springer-Verlag. doi:10.1007/978-3-642-00619-7_7
- Ruciński, M., Izzo, D., & Biscani, F. (2010). On the impact of the migration topology on the Island Model. *Parallel Computing*, 36(10-11), 555-571. doi:10.1016/j.parco.2010.04.002
- Runarsson, T. P. (2000). Stochastic ranking for constrained evolutionary optimization. *Evolutionary Computation, IEEE*, 4(3), 284-294. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=873238
- Runarsson, T. P. (2006). Approximate Evolution Strategy using Stochastic Ranking. *2006 IEEE International Conference on Evolutionary Computation* (pp. 745-752). IEEE. doi:10.1109/CEC.2006.1688386
- Sakai, S., & Takahama, T. (2007). Constrained Optimization by Applying a Stochastic Constrained Method to Particle Swarm Optimization. In S. Hiraki & M. Sakaguchi (Eds.), *System Sciences for Economics and Informatics* (pp. 71-87). Kyushu University Press.
- Singh, H. K., Isaacs, A., Nguyen, T. T., & Ray, T. (2009). Performance of infeasibility driven evolutionary algorithm (IDEA) on constrained dynamic single objective optimization problems. *2009 IEEE Congress on Evolutionary Computation* (pp. 3127-3134). IEEE. doi:10.1109/CEC.2009.4983339
- Singh, H. K., Ray, T., & Smith, W. (2010). Performance of infeasibility empowered memetic algorithm for CEC 2010 constrained optimization problems. *IEEE Congress on Evolutionary Computation* (pp. 1-8). IEEE. doi:10.1109/CEC.2010.5585946
- Sinha, A., Srinivasan, A., & Deb, K. (2006). A Population-Based, Parent Centric Procedure for Constrained Real-Parameter Optimization. *2006 IEEE International Conference on Evolutionary Computation* (pp. 239-245).
- Skolicki, Z. (2005). *An analysis of island models in evolutionary computation. Proceedings of the 2005 workshops on Genetic and evolutionary computation - GECCO '05*. ACM Press, New York, New York, USA.
- Skolicki, Z. M. (2007). *An Analysis of Island Models in Evolutionary Computation*. George Mason University.
- Smith, A. E., & Tate, D. M. (1993). Genetic Optimization Using a Penalty Function. In S. Forrest (Ed.), *Proc. of the Fifth International Conference on genetic Algorithms* (pp. 499-503). Morgan Kaufmann.
- Srinivasa, K. G., Venugopal, K. R., & Patnaik, L. M. (2007). A self-adaptive migration model genetic algorithm for data mining applications. *Information Sciences*, 177(20), 4295-4313. doi:10.1016/j.ins.2007.05.008
- Takahama, T., & Sakai, S. (2006). Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites. *2006 IEEE International Conference on Evolutionary Computation* (pp. 1-8).
- Takahama, T., & Sakai, S. (2010). Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based

mutation. *IEEE Congress on Evolutionary Computation* (pp. 1-9). IEEE. doi:10.1109/CEC.2010.5586484

Takahama, T., Sakai, S., & Iwane, N. (2005). Constrained optimization by the epsilon constrained hybrid algorithm of particle swarm optimization and genetic algorithm. *Proc. of the 18th Australian Joint Conference on Artificial Intelligence* (Vol. 3809, pp. 389–400). Sydney: Springer-Verlag.

Tasgetiren, M. F., Suganthan, P. N., Pan, Q.-K., Mallipeddi, R., & Sarman, S. (2010). An ensemble of differential evolution algorithms for constrained function optimization. *IEEE Congress on Evolutionary Computation* (pp. 1-8). IEEE. doi:10.1109/CEC.2010.5586396

Vrugt, J. A., Robinson, B. A., & Hyman, J. M. (2009). Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces. *IEEE Transactions on Evolutionary Computation*, 13(2), 243-259. doi:10.1109/TEVC.2008.924428

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.

Yang, Y. W., Xu, J. F., & Soh, C. K. (2006). An evolutionary programming algorithm for continuous global optimization. *European Journal of Operational Research*, 168(2), 354-369. doi:10.1016/j.ejor.2004.05.007

Yu, X., & Gen, M. (2010). *Introduction to Evolutionary Algorithms*. London: Springer. doi:0.1007/978-1-84996-129-5

Zielinski, K., & Laur, R. (2006a). Constrained Single-Objective Optimization Using Particle Swarm Optimization. *2006 IEEE International Conference on Evolutionary Computation* (pp. 443-450). IEEE. doi:10.1109/CEC.2006.1688343

Zielinski, K., & Laur, R. (2006b). Constrained Single-Objective Optimization Using Differential Evolution. *2006 IEEE International Conference on Evolutionary Computation* (pp. 223-230). Ieee. doi:10.1109/CEC.2006.1688312

de la Ossa, L., Gámez, J. a., & Puerta, J. M. (2006). Initial approaches to the application of islands-based parallel EDAs in continuous domains. *Journal of Parallel and Distributed Computing*, 66(8), 991-1001. doi:10.1016/j.jpdc.2006.03.005