Ecology-inspired Evolutionary Algorithm using Feasibility-based Grouping for Constrained Optimization

Ming Yuchi **Jong-Hwan Kim**
Dept. of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, Korea 305-701
{ycm, johkim}@rit.kaist.ac.kr

Abstract- When evolutionary algorithms are used for solving numerical constrained optimization problems, how to deal with the relationship between feasible and infeasible individuals can directly influence the final results. This paper proposes a novel ecology-inspired EA to balance the relationship between feasible and infeasible individuals. According to the feasibility of the individuals, the population is divided into two groups, feasible group and infeasible group. The evaluation and ranking of these two groups are performed separately. The number of parents from feasible group has a sigmoid relation with the number of feasible individuals, which is inspired by the ecological population growth in a confined space. The proposed method is tested using (μ, λ) evolution strategies with 13 benchmark problems. Experimental results show that the proposed method is capable of improving performance of the dynamic penalty method for constrained optimization problems.

1 Introduction

The general constrained optimization problem (P) is to find $\vec{x}$ so as to

$$\min_{\vec{x}} f(\vec{x}), \quad \vec{x} = (x_1, \dots x_n) \in R^n \tag{1}$$

where $\vec{x} \in F \subseteq S$. The *objective function f* is defined on the *search space* $S \subseteq R^n$ and the set $F \subseteq S$ defines the *feasible region*. Usually, the search space S is defined as an n-dimensional rectangle in R^n (domains of variables defined by their lower and upper bounds):

$$l(j) \leq x_j \leq u(j), \;\; j = 1, \dots, n \tag{2}$$

where the feasible region $F \subseteq S$ is defined by a set of m additional constraints ($m \geq 0$):

$$g_k(\vec{x}) \leq 0, \qquad k = 1, \dots, l, \tag{3}$$

$$h_k(\vec{x}) = 0, \quad k = l+1, \dots, m. \tag{4}$$

Any point $\vec{x} \subseteq F$ is called a feasible solution, otherwise, $\vec{x}$ is an infeasible solution.

When evolutionary algorithms are adopted to solve constrained optimization problems, intuitively, feasible individuals could be thought to have better fitness values than infeasible individuals (here "better" means to have more chance to survive and reproduce). Since all constrains of feasible individuals have already been satisfied, the only aim left is to find $\vec{x}$ minimize $f(\vec{x})$. Most of the existing evolutionary algorithms for constrained optimization problems follow such an idea and, more or less, underrated the importance of the infeasible individuals. For example, nearly all of the penalty methods add some "penalties" to the fitness functions of the infeasible individuals, and then rank the infeasible individuals with feasible individuals together [1], [2], [3]. Some other methods [4] directly assume that feasible individuals are always fitter than infeasible ones.

However, this kind of view ignores one important thing that evolutionary algorithm is a probabilistic and recurrent method. It is possible that some of the infeasible individuals carry more useful information than feasible individuals during evolution process. Moreover, quite often the system can reach the optimal point more easily if it is possible to "cross" an infeasible region (especially in non-convex feasible search space).

There have been some work on adjusting the relationship between feasible and infeasible individuals. In [5], [6], feasible and infeasible individuals were evaluated with different criteria. Other method like GENOCOP III [7] repaired the infeasible individuals for evaluation. However, for each particular problem, a specific repair strategy needs to be designed.

In this paper, based on the idea of fully utilizing the useful information of infeasible individuals, a novel evolutionary algorithm using feasibility-based grouping (EA_FG) is proposed for constrained optimization problems. In each generation, according to the feasibility of the individuals, the whole population is divided into two groups: feasible group and infeasible group. Evaluation and ranking of these two groups are performed in parallel and separately. The best individuals from feasible and infeasible groups are selected together as parents. The number of feasible parents has a sigmoid-type relation with that of feasible individuals, which is inspired by the natural ecological population growth in a confined space.

Any existing evolutionary algorithms for constrained optimization problems, which evaluate and rank feasible and infeasible individuals together, can be incorporated into EA_FG to improve the performance. In this paper, a dynamic penalty method is incorporated into EA_FG to test the effectiveness of EA_FG. The initial study on the EA_FG can be found in [8], [9], [10].

This paper is organized as follows. Section II presents a detailed description of the overall structure of EA_FG. Also, an ecology-inspired parent selection mechanism is discussed. In Section III, experimental results on 13 benchmark problems are presented and compared with a dynamic penalty method. Finally, Section IV concludes with some remarks.

2 EA_FG for Constrained Optimization

2.1 Overall Structure of EA_FG

Fig. 1 shows the flowchart of EA_FG. In the beginning of every generation, the whole population is divided into two groups: feasible group and infeasible group according to the feasibility of every individual. Then, the evaluation and ranking of these two groups are performed in parallel and separately. In order to share information of these two groups, the best feasible and infeasible individuals are selected as parent population. Parent population reproduces and generates offspring population. The offspring population is to be divided into feasible and infeasible groups again. The iteration keeps on working until the termination condition is satisfied.

When EA_FG is used to solve constrained optimization problems, two aspects should be noted. The first one is how to perform evaluation and ranking of infeasible groups. The second one is how to select parents from feasible and infeasible groups, in another words, how to decide the number of feasible parents and the number of infeasible parents. For the first aspect, any existing evaluation and ranking method for the whole population, can be adopted to evaluate and rank the infeasible group. In this paper, the dynamic penalty method [1] will be tried. For the second aspect, a novel method inspired by the natural population growth in a confined space will be used.

2.2 Ecological-inspired Parent Selection

In nature, the growth of a simple population in a confined space, where resources are not unlimited, is simply described by a graph that always looks sigmoid (Fig. 2(a)) [11]. In the early stage, resources are abundant, the death rate is minimal and the reproduction can take place as fast as possible. The population increases geometrically until an upper limit is approached. This upper limit, or saturation value is a constant for a particular set of conditions in a par-

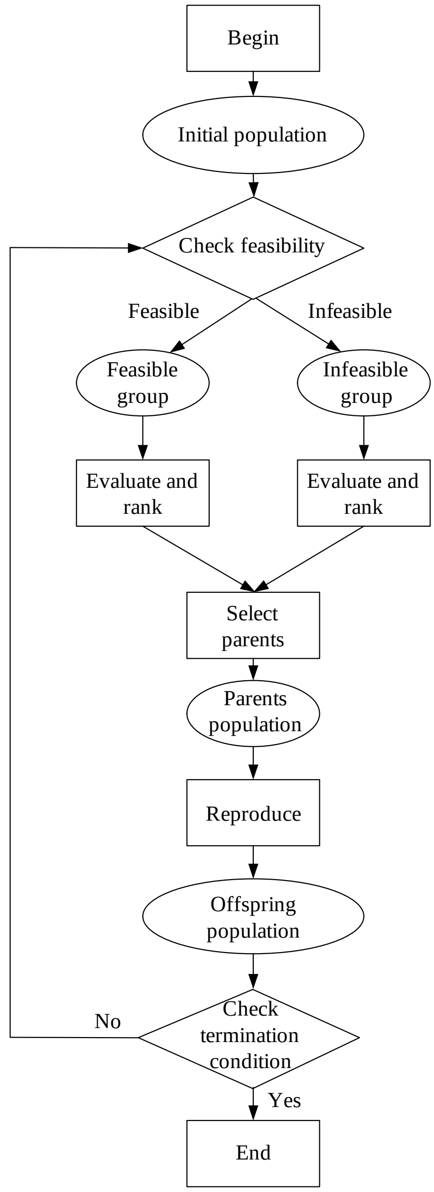

Figure 1: Flow chart of EA_FG

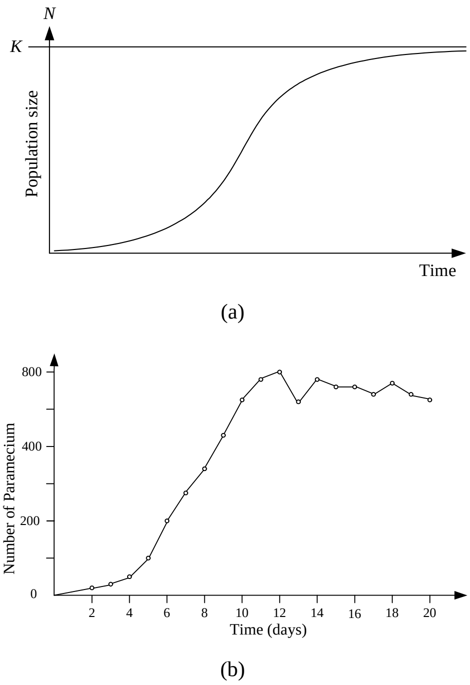

Figure 2: (a) Sigmoid population growth curve; (b) observed paramecium population growth.

ticular habitat and is called the carrying capacity (*k*). The population growth rate declines to zero as the population becomes more crowded and the population size stabilizes at the maximum that the environment can support, reaching an equilibrium population density. Fig. 2(b) shows one observed example: paramecium population growth.

Inspired by this phenomenon, a novel parent selection strategy is generated for EA_FG. The number of feasible parents which are selected from the feasible group and the number of feasible individuals will follow a sigmoid relation as follows:

i) $numFeaPar = ceil(sig(numFeaInd)),$ (5)

If $numFeaPar > numPar, numFeaPar = numPar.$ (6)

ii) $numInfeaPar = numPar - numFeaPar.$ (7)

where $numFeaPar$ represents 'number of feasible parents,' $numFeaInd$ 'number of feasible individuals,' $numPar$ 'number of parents,' $numInfeaPar$ 'number of infeasible parents.' sig(x) denotes a sigmoid-type equation, ceil(x) rounds the elements of x to the nearest integers towards infinity. Note that (6) restricts $numFeaPar$ not to exceed the predefined number of parents ($numPar$). Once numbers of feasible and infeasible parents are decided, the corresponding numbers of best individuals are selected from the feasible and infeasible groups, respectively, according to their ranking.

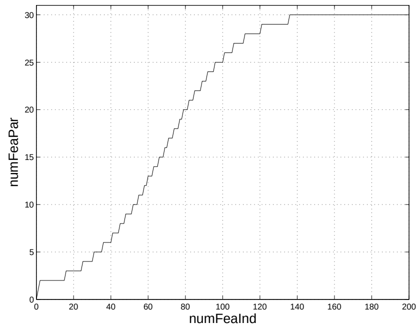

Figure 3: The sigmoid-type relation between the number of feasible parents (numFeaPar) and the number of feasible individuals (numFeaInd) of EA_FG with (30, 200)-ES.

To explain this strategy more clearly, assume that (30, 200)-ES is used for computation. Fig. 3 shows an example of the sigmoid relation between the number of feasible parents and the number of feasible individuals. (5) can be calculated by the following equation:

$$numFeaPar = ceil(\frac{30}{1 + 30 * exp(-0.05 * numFeaInd)}), \quad (8)$$

where 30 is the limit of $numFeaPar$, and -0.05 is set to make the curve similar to a natural population growth curve like Fig. 2(b). If $numFeaInd = 20$, (8) gives $numFeaPar = 3$. And by (7), $numInfeaPar = 30 - 3 = 27$. Therefore, the best 3 individuals are selected from the feasible group and best 27 individuals from the infeasible group, thus to form 30 parents for reproduction. Also, if $numFeaInd = 180$, (8) gives $numFeaPar = 30$. And by (7), $numInfeaPar = 30 - 30 = 0$. In this case, all 30 parents are from the feasible group.

3 Experimental Studies

Since EA_FG is an open structure algorithm, any evolutionary algorithm for numerical constrained optimization problems can be employed for the infeasible group. In this section, a dynamic penalty method was adopted to rank and evaluate the infeasible group of EA_FG, which would be called 'EA_FG_D' in the later part of this paper. EA_FG_D was tested and the results were compared with those of the dynamic penalty method in [12] on 13 benchmark functions. The details of these functions are listed in Appendix. Problems $G2$, $G3$, $G8$ and $G12$ are maximization problems. They were transformed into minimization problems using $-f(\vec{x})$. Problems $G3$, $G5$, $G11$ and $G13$ include one or several equality constraints. All of these equality constraints were converted into inequality constraints, $|h(\vec{x})| - \delta \leq 0$, using the degree of violation $\delta = 0.0001$.

Table 1: Experimental Dynamic Penalty Method; "-" Means no Feasible Solutions Were Found, Data are From [12].

Function	optimal	best	median	worst	mean	st. dev.	g_m
$G1$	-15.000	-15.000	-15.000	-15.000	-15.000	$7.9E-005$	217
$G2$	-0.803619	-0.803587	-0.785907	-0.751624	-0.784868	$1.5E-002$	1235
$G3$	-1.000	-0.583	-0.045	-0.001	-0.103	$1.4E-001$	996
$G4$	-30665.539	-30365.488	-30060.607	-29871.442	-30072.458	$1.2E+002$	4
$G5$	5126.498	$-$	$-$	$-$	$-$	$-$	$-$
$G6$	-6961.814	-6911.247	-6547.354	-5868.028	-6540.012	$2.6E+002$	13
$G7$	24.306	24.309	24.375	25.534	24.421	$2.2E-001$	180
$G8$	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	$2.8E-017$	421
$G9$	680.630	680.632	680.648	680.775	680.659	$3.2E-002$	1739
$G10$	7049.331	$-$	$-$	$-$	$-$	$-$	$-$
$G11$	0.750	0.750	0.750	0.750	0.750	$9.1E-006$	61
$G12$	-1.000000	-1.000000	-0.999818	-0.999573	-0.999838	$1.3E-004$	68
$G13$	0.053950	0.514152	0.996674	0.998156	0.965397	$9.4E-002$	1750

A (μ, λ)-ES was employed for recombination and mutation. For impartial comparison, all parameters of the ES used here were the same as those of [12]. For each of the benchmark problems, 30 independent runs were performed using $(30, 200)$-ES. The termination condition was set to be 1,750 generations. The initial population of $\vec{x}$ was generated according to a uniform n-dimensional probability distribution over the search space S. The ecology-inspired parent selection in Fig. 3 was adopted.

Table 1 and Table 2 show the simulation results with a dynamic penalty method and EA_FG_D, respectively. The median number of generations for finding the best solution in each run is indicated by g_m in the tables. The tables also show the known 'optimal' solution for each problem and statistics for the 30 independent runs. Best, median, worst and mean values of 30 runs were used as performance criteria. 13 problems totally have $13 \times 4 = 52$ criteria. Among the 52 criteria,

- EA_FG_D performed better on 27 criteria,
- EA_FG_D performed worse on 6 criteria, and
- EA_FG_D performed the same on 19 criteria.

For problems $G1$, $G8$ and $G11$, both algorithms performed well and found the optimal solutions for all 30 runs. For problem $G10$, both algorithms failed to find the optimal solution. For problem $G2$, EA_FG_D provided 'similar' results to the dynamic penalty method. It performed better on *worst* and *mean*, but worse on *best* and *median* than the dynamic penalty method. For the rest of problems, EA_FG_D outperformed the dynamic penalty method except problem $G13$. For problem $G3$, *best* of the dynamic penalty method was -0.583, while *best* of EA_FG_D could reach the optimal value -1.000. For problems $G4$, $G6$ and $G9$, EA_FG_D performed significantly better in terms of all four criteria. It should be noted that for problem $G5$, EA_FG_D could find a feasible solution 3 times out of 30 runs, while the dynamic penalty method failed to find the solution.

4 Conclusion

This paper proposed a new constraint handling technique: ecology-inspired evolutionary algorithm using feasibility-based grouping. This method divides the population into two groups, feasible group and infeasible group according to the feasibility of the individuals. The evaluation and ranking of these two groups are performed in parallel and separately. The number of feasible parents has a sigmoid-type relation with that of feasible individuals which is inspired by the ecological population growth in a confined space in nature. In addition, a dynamic penalty method was modified and included into EA_FG_D to evaluate and rank the infeasible group. EA_FG_D was tested on a set of 13 benchmark problems. Experimental results showed EA_FG_D could improve the performance of the dynamic penalty method on most problems.

5 Acknowledgments

This work was supported by the **ITRC-IRRC** (Intelligent Robot Research Center) of the Korea Ministry of Information and Communication in 2004.

Bibliography

[1] J. Joines and C. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs," in *Proc. 1st IEEE Int. Conf. Evolutionary Computation*, IEEE Press, 1994, pp. 579-584.

[2] Z. Michalewicz and N. Attia, "Evolutionary optimization of constrained problems," in *Proc. 3rd Annu. Conf. Genetic Algorithms*, World scientific, 1994, pp. 98-108.

[3] S. B. Hamida and M. Schoenauer, "ASCHEA: new results using adaptive segregational constraint handling,"

Table 2: Experimental Results of EA_FG_D; The Subscript 4 in the Function Name $G5_{(3)}$ Indicates 3 Feasible Solutions Found Among 30 tests; "-" Means no Feasible Solutions Were Found.

Function	optimal	best	median	worst	mean	st. dev.	g_m
$G1$	-15.000	-15.000	-15.000	-15.000	-15.000	$6.3E-006$	676
$G2$	-0.803619	-0.803567	-0.785643	-0.767080	-0.787585	$1.1E-002$	1110
$G3$	-1.000	-1.000	-0.088	-0.001	-0.121	$1.8E-001$	1743
$G4$	-30665.539	-30665.457	-30561.593	-30185.263	-30540.911	$1.2E+002$	545
$G5_{(3)}$	5126.498	5208.553	5377.300	5462.911	5349.588	$1.1E+002$	1267
$G6$	-6961.814	-6961.062	-6651.928	-6179.194	-6652.052	$2.2E+002$	13
$G7$	24.306	24.309	24.335	24.649	24.378	$9.0E-002$	437
$G8$	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	$0.0E+000$	369
$G9$	680.630	680.630	680.637	680.677	680.643	$1.3E-002$	1632
$G10$	7049.331	–	–	–	–	–	–
$G11$	0.750	0.750	0.750	0.750	0.750	$2.2E-007$	610
$G12$	-1.000000	-1.000000	-0.999970	-0.999905	-0.999968	$2.3E-005$	648
$G13$	0.053950	0.742678	0.997744	0.998500	0.984344	$4.7E-002$	1749

in *Proc. Cong. Evolutionary Computation*, 2002, vol. 1, pp. 884-889.

[4] S. B. Hamida and A. Petrowski "The need for improving the exploration operators for constrained optimization problems," in *Proc. Cong. Evolutionary Computation*, 2000, vol. 2, pp. 1176-1183.

[5] R. Hinterding and Z. Michalewicz, "Your brains and my beauty: parent matching for constrained optimisation," in *Proc. 5th IEEE Int. Conf. Evolutionary Computation*, IEEE Press, 1998, pp. 810-815.

[6] F. Hoffmeister and J. Sprave, "Problem-independent handling of constraints by use of metric penalty functions," in *Proc. 5th Annu. Conf. Evolutionary Programming*, M.I.T Press, 1996, pp. 289-294.

[7] Z. Michalewicz and G. Nazhiyath, "Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints," in *Proc. IEEE Conf. Evolutionary Computation*, IEEE Press, 1995, pp. 647 - 651.

[8] M. Yuchi and J.-H. Kim, "A grouping-based evolutionary algorithm for constrained optimization problem," in *Proc. Cong. Evolutionary Computation*, 2003, vol. 3, pp. 1507-1512.

[9] M. Yuchi and J.-H. Kim, "Grouping-based evolutionary algorithm: seeking balance between feasible and infeasible individuals of constrained optimization problems," in *Proc. Cong. Evolutionary Computation*, 2004, vol. 1, pp. 280-287.

[10] M. Yuchi and J.-H. Kim, "Grouping-based evolutionary algorithm improves the performance of dynamic penalty method for constrained optimization problems," in *Proc. 5th Intl. Conf. Simulated Evolution and Learning*, 2005.

[11] A. Machenzie, A. S. Ball, and S. R. Virdee, *Instant Notes in Ecology*, Springer-Verlag, Singapore, 1998.

[12] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284-294, 2000.

A Test Function Suite

Problem $G1$: Minimize

$$f(\vec{x}) = 5\sum_{j=1}^{4} x_j - 5\sum_{j=1}^{4} x_j^2 - \sum_{j=5}^{13} x_j,$$

subject to

$$\begin{aligned}
g_1(\vec{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0,\\
g_2(\vec{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0,\\
g_3(\vec{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0,\\
g_4(\vec{x}) &= -8x_1 + x_{10} \le 0,\\
g_5(\vec{x}) &= -8x_2 + x_{11} \le 0,\\
g_6(\vec{x}) &= -8x_3 + x_{12} \le 0,\\
g_7(\vec{x}) &= -2x_4 - x_5 + x_{10} \le 0,\\
g_8(\vec{x}) &= -2x_6 - x_7 + x_{11} \le 0,\\
g_9(\vec{x}) &= -2x_8 - x_9 + x_{12} \le 0,
\end{aligned}$$

and bounds

$$\begin{aligned}
&0 \le x_j \le 1 \; (j = 1, \ldots, 9),\\
&0 \le x_j \le 100 \; (j = 10, 11, 12), 0 \le x_{13} \le 1.
\end{aligned}$$

The global minimum is at $\vec{x}^*$=$(1, 1, 1, 1$,1,$1, 1, 1, 1, 3, 3, 3, 1)$, and $f(\vec{x}^*) = -15$.

Problem $G2$: Maximize

$$f(\vec{x}) = \left| \frac{\sum_{j=1}^{n} \cos^4(x_j) - 2\prod_{j=1}^{n} \cos^2(x_j)}{\sqrt{\sum_{j=1}^{n} j x_j^2}} \right|,$$

subject to

$$g_1(\vec{x}) = 0.75 - \prod_{j=1}^{n} x_j \le 0,$$

$$g_2(\vec{x}) = \sum_{j=1}^{n} x_j - 7.5n \le 0,$$

and bounds

$$0 \le x_j \le 10 \; (j = 1, \ldots, n),$$

where $n = 20$. The global maximum is unknown; the known solution is $f(\vec{x}^*) = 0.803619$.

Problem $G3$: Maximize

$$f(\vec{x}) = (\sqrt{n})^n \prod_{j=1}^{n} x_j,$$

subject to

$$h_1(\vec{x}) = \sum_{j=1}^{n} x_j^2 - 1 = 0,$$

and bounds

$$0 \le x_j \le 1 \; (j = 1, \ldots, n),$$

where $n = 10$. The global minimum is at $x_j^* = 1/\sqrt{n}$ $(j = 1, \ldots, n)$, and $f(\vec{x}^*) = 1$.

Problem $G4$: Minimize

$$f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141,$$

subject to

$$\begin{aligned}
g_1(\vec{x}) =& 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 \\
&- 0.0022053x_3x_5 - 92 \le 0, \\
g_2(\vec{x}) =& -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 \\
&+ 0.0022053x_3x_5 \le 0, \\
g_3(\vec{x}) =& 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 \\
&+ 0.0021813x_3^2 - 110 \le 0, \\
g_4(\vec{x}) =& -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 \\
&- 0.0021813x_3^2 + 90 \le 0, \\
g_5(\vec{x}) =& 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 \\
&+ 0.0019085x_3x_4 - 25 \le 0, \\
g_6(\vec{x}) =& -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 \\
&- 0.0019085x_3x_4 + 20 \le 0,
\end{aligned}$$

and bounds

$$78 \le x_1 \le 102, 33 \le x_2 \le 45, 27 \le x_j \le 45, \; (j = 3, 4, 5).$$

The optimal solution is at $\vec{x}^*$=$(78, 33, 29.995256025682, 45, 36.775812905788)$, and $f(\vec{x}^*) = -30665.539$.

Problem $G5$: Minimize

$$f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3,$$

subject to

$$\begin{aligned}
g_1(\vec{x}) &= -x_4 + x_3 - 0.55 \le 0, \\
g_2(\vec{x}) &= -x_3 + x_4 - 0.55 \le 0, \\
h_1(\vec{x}) &= 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) \\
&\quad + 894.8 - x_1 = 0, \\
h_2(\vec{x}) &= 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) \\
&\quad + 894.8 - x_2 = 0, \\
h_3(\vec{x}) &= 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) \\
&\quad + 1294.8 = 0,
\end{aligned}$$

and bounds

$$\begin{aligned}
&0 \le x_1 \le 1200, 0 \le x_2 \le 1200, \\
&-0.55 \le x_3 \le 0.55, -0.55 \le x_4 \le 0.55.
\end{aligned}$$

The best know solution is at $\vec{x}^*$=$(679.9453, 1026.067, 0.1188764, -0.3962336)$, and $f(\vec{x}^*) = 5126.4981$.

Problem $G6$: Minimize

$$f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3,$$

subject to

$$\begin{aligned}
g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0, \\
g_2(\vec{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0,
\end{aligned}$$

and bounds

$$13 \le x_1 \le 100, 0 \le x_2 \le 100.$$

The known global solution is $\vec{x}^* = (14.095, 0.84296)$, and $f(\vec{x}^*) = -6961.81388$.

Problem $G7$: Minimize

$$\begin{aligned}
f(\vec{x}) =& x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + \\
&4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + \\
&7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45,
\end{aligned}$$

subject to

$$\begin{aligned}
g_1(\vec{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0, \\
g_2(\vec{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0, \\
g_3(\vec{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0, \\
g_4(\vec{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0, \\
g_5(\vec{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0, \\
g_6(\vec{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0, \\
g_7(\vec{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0, \\
g_8(\vec{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,
\end{aligned}$$

and bounds

$$-10 \le x_j \le 10 \; (j = 1, \ldots, 10).$$

The optimal solution is at $\vec{x}^*$=$(2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$, and $f(\vec{x}^*) = 24.3062091$.

Problem $G8$: Maximize

$$f(\vec{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)},$$

subject to

$$g_1(\vec{x}) = x_1^2 - x_2 + 1 \le 0,$$
$$g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \le 0,$$

and bounds

$$-10 \le x_1 \le 10, -10 \le x_2 \le 10, .$$

The optimal solution is at $\vec{x}^* = (1.2279713, 4.2453733)$, and $f(\vec{x}^*) = 0.095825$.

Problem $G9$: Minimize

$$f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7,$$

subject to

$$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0,$$
$$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0,$$
$$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0,$$
$$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0,$$

and bounds

$$-10 \le x_j \le 10 \ (j = 1, \ldots, 7).$$

The known global solution is at $\vec{x}^*$= (2.330499,1.951372, -0.4775414,4.365726,-0.6244870, 1.038131,1.594227), and $f(\vec{x}^*) = 680.6300573$.

Problem $G10$: Minimize

$$f(\vec{x}) = x_1 + x_2 + x_3,$$

subject to

$$g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \le 0,$$
$$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0,$$
$$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \le 0,$$
$$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0,$$
$$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0,$$
$$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0,$$

and bounds

$$100 \le x_1 \le 10000, 1000 \le x_j \le 10000 \ (j = 2, 3),$$
$$10 \le x_j \le 1000 \ (j = 4, \ldots, 8).$$

The optimal solution is at $\vec{x}^*$=$(579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$, and $f(\vec{x}^*) = 7049.3307$.

Problem $G11$: Minimize

$$f(\vec{x}) = x_1^2 + (x_2 - 1)^2,$$

subject to

$$h_1(\vec{x}) = x_2 - x_1^2 = 0,$$

and bounds

$$-1 \le x_1 \le 1, -1 \le x_2 \le 1.$$

The optimal solution is at $\vec{x}^* = (\pm 1/\sqrt{2}, 1/\sqrt{2})$, and $f(\vec{x}^*) = 0.75$.

Problem $G12$: Maximize

$$f(\vec{x}) = (100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100,$$

subject to

$$g(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \le 0,$$

and bounds

$$0 \le x_j \le 10 \ (j = 1, 2, 3),$$

where $p, q, r = 1, 2, \ldots, 9$. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exists p, q, r such that the above inequality holds. The optimal solution is at $\vec{x}^* = (5, 5, 5)$, and $f(\vec{x}^*) = 1$.

Problem $G13$: Minimize

$$f(\vec{x}) = e^{x_1x_2x_3x_4x_5},$$

subject to

$$h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,$$
$$h_2(\vec{x}) = x_2x_3 - 5x_4x_5 = 0,$$
$$h_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0,$$

and bounds

$$-2.3 \le x_j \le 2.3 \ (j = 1, 2), -3.2 \le x_j \le 3.2 \ (j = 3, 4, 5).$$

The optimal solution is at $\vec{x}^*$=$(-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$, and $f(\vec{x}^*) = 0.0539498$.