A Hybrid Genetic Pattern Search Augmented Lagrangian Method for Constrained Global Optimization

Lino Costa, Isabel Espírito Santo and Edite M.G.P. Fernandes

Production and Systems Department,
School of engineering,
University of Minho, Portugal.
{lac,iapinho,emgpf}@dps.uminho.pt

14th International Congress on Computational and Applied Mathematics (ICCAM 2009), Antalya, Turkey

September 29 and October 2, 2009
Outline

1. Motivation

2. Augmented Lagrangian Technique

3. Genetic Algorithm (GA)

4. Hooke and Jeeves (HJ) Pattern Search

5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm

6. Numerical results

7. Conclusions and Future Work
Motivation

Augmented Lagrangian Technique

Genetic Algorithm (GA)

Hooke and Jeeves (HJ) Pattern Search

Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm

Numerical results

Conclusions and Future Work
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Outline

1 Motivation
2 Augmented Lagrangian Technique
3 Genetic Algorithm (GA)
4 Hooke and Jeeves (HJ) Pattern Search
5 Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6 Numerical results
7 Conclusions and Future Work
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Outline

1 Motivation

2 Augmented Lagrangian Technique

3 Genetic Algorithm (GA)

4 Hooke and Jeeves (HJ) Pattern Search

5 Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm

6 Numerical results

7 Conclusions and Future Work
Constrained Global Optimization

The following problem is under consideration

\[
\begin{align*}
\min_{x \in \Omega} & \quad f(x) \\
\text{s.t.} & \quad g_j(x) \leq 0 \quad j = 1, \ldots, p \\
& \quad b_i(x) = 0 \quad i = 1, \ldots, m
\end{align*}
\]

(inequality constraints)

(equality constraints)

where \(x\) is an \(n\) dimensional vector and \(\Omega \subset \mathbb{R}^n\)

\((\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\})\).

Global Optimization

We aim to compute an approximation to the global optimum of the constrained problem.

Assumptions

\(f(x)\) is not assumed to be convex thus many local minima may exist.
Constrained Global Optimization

The following problem is under consideration

\[
\min_{x \in \Omega} \quad f(x) \quad \text{(objective function)}
\]
\[
s.t. \quad g_j(x) \leq 0 \quad j = 1, \ldots, p \quad \text{(inequality constraints)}
\]
\[
b_i(x) = 0 \quad i = 1, \ldots, m \quad \text{(equality constraints)}
\]

where \( x \) is an \( n \) dimensional vector and \( \Omega \subset \mathbb{R}^n \)
\( (\Omega = \{ x \in \mathbb{R}^n : l \leq x \leq u \} ) \).

Global Optimization

We aim to compute an approximation to the global optimum of the constrained problem.

Assumptions

\( f(x) \) is not assumed to be convex thus many local minima may exist.
Constrained Global Optimization

The following problem is under consideration

\[
\begin{align*}
\min_{x \in \Omega} & \quad f(x) \quad \text{(objective function)} \\
\text{s.t.} & \quad g_j(x) \leq 0 \quad j = 1, \ldots, p \quad \text{(inequality constraints)} \\
& \quad b_i(x) = 0 \quad i = 1, \ldots, m \quad \text{(equality constraints)}
\end{align*}
\]

where \( x \) is an \( n \) dimensional vector and \( \Omega \subset \mathbb{R}^n \)
\((\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\})

Global Optimization

We aim to compute an approximation to the global optimum of the constrained problem.

Assumptions

\( f(x) \) is not assumed to be convex thus many local minima may exist.
Motivation

Why global optimization techniques?
- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?
- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?

- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?

- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?

- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?

- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?

- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?

- Overall successful and efficient general solvers do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?

- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?

- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?
- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?
- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?

- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?

- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?
- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?
- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Motivation

Why global optimization techniques?
- non-convex problems;
- many local minima;
- no differentiability conditions;
- a derivative-free method for global optimization must be used.

Why hybridization?
- Overall successful and efficient general solver do not exist;
- Stochastic algorithms, e.g., genetic algorithms are good at identifying promising areas of the search space (exploration), but less good at fine-tuning approximations to the minimum (exploitation);
- Local search algorithms, e.g., Pattern Search are good at improving approximations to the minimum (exploitation) but less good at exploring search space.
Outline

1 Motivation

2 Augmented Lagrangian Technique

3 Genetic Algorithm (GA)

4 Hooke and Jeeves (HJ) Pattern Search

5 Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm

6 Numerical results

7 Conclusions and Future Work
Augmented Lagrangian Technique

What is an augmented Lagrangian technique?

- The solution of the constrained optimization problem is obtained by solving a sequence of simple subproblems;
- the objective function of the subproblem incorporates the equality and inequality constraints;
- however, the simple bounds are left explicit.

The subproblem objective (augmented Lagrangian) function:

\[
\Phi(x; \lambda, \delta, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i b_i(x) + \frac{1}{2\mu} \sum_{i=1}^{m} b_i(x)^2 \\
+ \frac{\mu}{2} \sum_{j=1}^{p} \left( \max \left(0, \delta_j + \frac{g_j(x)}{\mu} \right)^2 - \delta_j^2 \right)
\]

- \(\lambda\) - multiplier vector associated with equality constraints
- \(\delta\) - multiplier vector associated with inequality constraints
- \(\mu\) - penalty parameter

Penalty terms aim to penalize constraints violation.
Augmented Lagrangian Technique

What is an augmented Lagrangian technique?

- The solution of the constrained optimization problem is obtained by solving a sequence of simple subproblems;
- the objective function of the subproblem incorporates the equality and inequality constraints;
- however, the simple bounds are left explicit.

The subproblem objective (augmented Lagrangian) function:

\[
\Phi(x; \lambda, \delta, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i b_i(x) + \frac{1}{2\mu} \sum_{i=1}^{m} b_i(x)^2 \\
+ \frac{\mu}{2} \sum_{j=1}^{p} \left( \max \left( 0, \delta_j + \frac{g_j(x)}{\mu} \right)^2 - \delta_j^2 \right)
\]

\(\lambda\) - multiplier vector associated with equality constraints
\(\delta\) - multiplier vector associated with inequality constraints
\(\mu\) - penalty parameter

Penalty terms aim to penalize constraints violation.
Augmented Lagrangian Technique

What is an augmented Lagrangian technique?

- The solution of the constrained optimization problem is obtained by solving a sequence of simple subproblems;
- The objective function of the subproblem incorporates the equality and inequality constraints;
- however, the simple bounds are left explicit.

The subproblem objective (augmented Lagrangian) function:

$$\Phi(x; \lambda, \delta, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i b_i(x) + \frac{1}{2\mu} \sum_{i=1}^{m} b_i(x)^2 + \frac{\mu}{2} \sum_{j=1}^{p} \left( \max \left(0, \delta_j + \frac{g_j(x)}{\mu} \right)^2 - \delta_j^2 \right)$$

$\lambda$ - multiplier vector associated with equality constraints
$\delta$ - multiplier vector associated with inequality constraints
$\mu$ - penalty parameter

Penalty terms aim to penalize constraints violation.
Augmented Lagrangian Technique

What is an augmented Lagrangian technique?

- The solution of the constrained optimization problem is obtained by solving a sequence of simple subproblems;
- the objective function of the subproblem incorporates the equality and inequality constraints;
- however, the simple bounds are left explicit.

The subproblem objective (augmented Lagrangian) function:

\[
\Phi(x; \lambda, \delta, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i b_i(x) + \frac{1}{2\mu} \sum_{i=1}^{m} b_i(x)^2 + \frac{\mu}{2} \sum_{j=1}^{p} \left( \max \left(0, \delta_j + \frac{g_j(x)}{\mu} \right)^2 - \delta_j^2 \right)
\]

\(\lambda\) - multiplier vector associated with equality constraints
\(\delta\) - multiplier vector associated with inequality constraints
\(\mu\) - penalty parameter

*Penalty terms aim to penalize constraints violation.*
Augmented Lagrangian Technique

Goal: to solve a sequence of subproblems:

For each $j$, and fixed $\lambda_j$, $\delta_j$ e $\mu_j$

$$\text{minimize} \quad \Phi_j(x) \equiv \Phi(x; \lambda_j, \delta_j, \mu_j)$$

subject to $l \leq x \leq u$,

As $j \to \infty$, $\mu_j \to 0$, and $x_{\text{min}}^j \to x_{\text{min}}$.

How to solve the subproblems?

Since the subproblems have non-differentiable functions, a combination of a Genetic Algorithm (GA) with Hooke and Jeeves (HJ) Pattern Search algorithm was developed: the Hybrid Genetic Pattern Search Augmented Lagrangian algorithm (GAPSAL).
Augmented Lagrangian Technique

Goal: to solve a sequence of subproblems:

For each \( j \), and fixed \( \lambda^j, \delta^j, \mu^j \)

\[
\begin{align*}
\text{minimize} & \quad \Phi^j(x) \equiv \Phi(x; \lambda^j, \delta^j, \mu^j) \\
\text{subject to} & \quad l \leq x \leq u,
\end{align*}
\]

As \( j \to \infty, \mu^j \to 0, \) and \( x^j_{\text{min}} \to x_{\text{min}}. \)

How to solve the subproblems?

Since the subproblems have non-differentiable functions, a combination of a Genetic Algorithm (GA) with Hooke and Jeeves (HJ) Pattern Search algorithm was developed: the Hybrid Genetic Pattern Search Augmented Lagrangian algorithm (GAPSAL).
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Algorithm framework

A genetic algorithm is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1999).

Outline

1. Randomly initialize an initial population (each individual of the population is a real vector representing the decision variables).
2. Evaluate the fitness of the individuals of the population.
3. Select a pool of individuals from the population according to their fitness by a tournament selection.
4. Generate a set of offspring obtained from individuals of the pool using crossover and mutation operators.
5. Verify the stopping criteria. If not met then goto step 3.
Genetic Algorithm (GA)

Algorithm framework

A genetic algorithm is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1999).

Outline

1. Randomly initialize an initial population (each individual of the population is a real vector representing the decision variables).
2. Evaluate the fitness of the individuals of the population.
3. Select a pool of individuals from the population according to their fitness by a tournament selection.
4. Generate a set of offspring obtained from individuals of the pool using crossover and mutation operators.
5. Verify the stopping criteria. If not met then goto step 3.
Algorithm framework

A genetic algorithm is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1999).

Outline

1. Randomly initialize an initial population (each individual of the population is a real vector representing the decision variables).
2. Evaluate the fitness of the individuals of the population.
3. Select a pool of individuals from the population according to their fitness by a tournament selection.
4. Generate a set of offspring obtained from individuals of the pool using crossover and mutation operators.
5. Verify the stopping criteria. If not met then goto step 3.
Algorithm framework

A genetic algorithm is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1999).

Outline

1. Randomly initialize an initial population (each individual of the population is a real vector representing the decision variables).
2. Evaluate the fitness of the individuals of the population.
3. Select a pool of individuals from the population according to their fitness by a tournament selection.
4. Generate a set of offspring obtained from individuals of the pool using crossover and mutation operators.
5. Verify the stopping criteria. If not met then goto step 3.
Algorithm framework

A genetic algorithm is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1999).

Outline

1. Randomly initialize an initial population (each individual of the population is a real vector representing the decision variables).
2. Evaluate the fitness of the individuals of the population.
3. Select a pool of individuals from the population according to their fitness by a tournament selection.
4. Generate a set of offspring obtained from individuals of the pool using crossover and mutation operators.
5. Verify the stopping criteria. If not met then goto step 3.
Algorithm framework

A genetic algorithm is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1999).

Outline

1. Randomly initialize an initial population (each individual of the population is a real vector representing the decision variables).
2. Evaluate the fitness of the individuals of the population.
3. Select a pool of individuals from the population according to their fitness by a tournament selection.
4. Generate a set of offspring obtained from individuals of the pool using crossover and mutation operators.
5. Verify the stopping criteria. If not met then goto step 3.
Algorithm framework

A genetic algorithm is a population based algorithm that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (Goldberg, 1999).

Outline

1. Randomly initialize an initial population (each individual of the population is a real vector representing the decision variables).
2. Evaluate the fitness of the individuals of the population.
3. Select a pool of individuals from the population according to their fitness by a tournament selection.
4. Generate a set of offspring obtained from individuals of the pool using crossover and mutation operators.
5. Verify the stopping criteria. If not met then goto step 3.
Fitness and Selection

**The fitness function**

The *fitness* function corresponds to the function of the subproblem defined by the Augmented Lagrangian technique.

**The Tournament Selection**

Tournaments are played between individuals and the better individual is chosen for the pool (survival of the fittest principle). The process is repeated until the pool is fulfilled (the size of the pool is inferior to the population size in order to implement elitism).
Fitness and Selection

The \textit{fitness} function

The \textit{fitness} function corresponds to the function of the subproblem defined by the Augmented Lagrangian technique.

The Tournament Selection

Tournaments are played between individuals and the better individual is chosen for the pool (survival of the fittest principle). The process is repeated until the pool is fulfilled (the size of the pool is inferior to the population size in order to implement elitism).
Genetic Operators

The *crossover* operator
Simulated Binary Crossover (SBX) that simulates the working principle of single-point crossover operator for binary strings. Two offspring are generated from two parents randomly selected from the pool (Deb, 1995).

The *mutation* operator
Polynomial mutation that guarantees that the probability of creating a point closer to the parent is more than the probability of creating one away from it.
Genetic Operators

The *crossover* operator

Simulated Binary Crossover (SBX) that simulates the working principle of single-point crossover operator for binary strings. Two offspring are generated from two parents randomly selected from the pool (Deb, 1995).

The *mutation* operator

Polynomial mutation that guarantees that the probability of creating a point closer to the parent is more than the probability of creating one away from it.
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Hooke and Jeeves pattern search method

In order to obtain a high accuracy approximation to the minimum, we implemented:

1. a derivative-free method, known as pattern search method, as outlined in Lewis & Torczon (1999);

The Hooke and Jeeves (HJ) method performs two types of moves:

- the exploratory move carries out a coordinate search – a search along the coordinate axes – about a selected iterate, with a step size $\Delta k$;
- when $x_k$ is a successful iterate, the pattern move – a promising direction – is defined by $x_k = x_{k-1}$. 

Lino Costa (UMinho - PT)
Hooke and Jeeves pattern search method

In order to obtain a high accuracy approximation to the minimum, we implemented:

1. a derivative-free method, known as **pattern search** method, as outlined in Lewis & Torczon (1999);


The Hooke and Jeeves (HJ) method performs two types of moves:

- the exploratory move carries out a coordinate search – a search along the coordinate axes – about a selected iterate, with a step size $\Delta_k$;
- when $x_k$ is a successful iterate, the pattern move – a promising direction – is defined by $x_k - x_{k-1}$.
Hooke and Jeeves pattern search method

In order to obtain a high accuracy approximation to the minimum, we implemented:

1. a derivative-free method, known as **pattern search** method, as outlined in Lewis & Torczon (1999);

The Hooke and Jeeves (HJ) method performs two types of moves:

- the exploratory move carries out a coordinate search - a search along the coordinate axes - about a selected iterate, with a step size $\Delta x$;
- when $x_k$ is a successful iterate, the pattern move - a promising direction - is defined by $x_{k+1} = x_k + \Delta x$. 

Hooke and Jeeves pattern search method

In order to obtain a high accuracy approximation to the minimum, we implemented:

1. a derivative-free method, known as **pattern search** method, as outlined in Lewis & Torczon (1999);

The Hooke and Jeeves (HJ) method performs two types of moves:

- the **exploratory move** carries out a coordinate search - a search along the coordinate axes - about a selected iterate, with a step size $\Delta_k$;
- when $x_k$ is a successful iterate, the **pattern move** - a promising direction - is defined by $x_k - x_{k-1}$. 
Hooke and Jeeves pattern search method

In order to obtain a high accuracy approximation to the minimum, we implemented:

1. a derivative-free method, known as **pattern search** method, as outlined in Lewis & Torczon (1999);

The Hooke and Jeeves (HJ) method performs two types of moves:

- the **exploratory move** carries out a coordinate search - a search along the coordinate axes - about a selected iterate, with a step size $\Delta_k$;
- when $x_k$ is a successful iterate, the **pattern move** - a promising direction - is defined by $x_k - x_{k-1}$.
Hooke and Jeeves pattern search method

In order to obtain a high accuracy approximation to the minimum, we implemented:

1. a derivative-free method, known as pattern search method, as outlined in Lewis & Torczon (1999);

The Hooke and Jeeves (HJ) method performs two types of moves:

- the exploratory move carries out a coordinate search - a search along the coordinate axes - about a selected iterate, with a step size $\Delta_k$;
- when $x_k$ is a successful iterate, the pattern move - a promising direction - is defined by $x_k - x_{k-1}$.
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi_j(x)$, are solved by GA and HJ.

**GAPSAL**

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   - $y^j \leftarrow \text{GA}(x^j)$
     (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA)
   - $x^{j+1} \leftarrow \text{HJ}(y^j)$
     (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ)
   - Update Augmented Lagrangian parameters
   - Set $j \leftarrow j + 1$
4. Set $x_{\text{min}} \leftarrow x^{j+1}$ as the best approximation found.
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi^j(x)$, are solved by GA and HJ.

**GAPSAL**

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   - $y^j \leftarrow \text{GA}(x^j)$
     (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA)
   - $x^{j+1} \leftarrow \text{HJ}(y^j)$
     (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ)
   - Update Augmented Lagrangian parameters
   - Set $j \leftarrow j + 1$
4. Set $x_{\text{min}} \leftarrow x^{j+1}$ as the best approximation found.
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi^j(x)$, are solved by GA and HJ.

**GAPSAL**

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   - $y^j \leftarrow GA(x^j)$
     (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA).
   - $x^{j+1} \leftarrow HJ(y^j)$
     (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ).
   - Update Augmented Lagrangian parameters.
   - Set $j \leftarrow j + 1$.
4. Set $x_{\text{min}} \leftarrow x^{j+1}$ as the best approximation found.
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi^j(x)$, are solved by GA and HJ.

GAPSAL

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   - $y^j \leftarrow GA(x^j)$
     (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA).
   - $x^{j+1} \leftarrow HJ(y^j)$
     (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ).
   - Update Augmented Lagrangian parameters.
   - Set $j \leftarrow j + 1$.
4. Set $x_{\text{min}} \leftarrow x^{j+1}$ as the best approximation found.
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi^j(x)$, are solved by GA and HJ.

**GAPSAL**

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   - $y^j \leftarrow GA(x^j)$
   - (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA).
   - $x^{j+1} \leftarrow HJ(y^j)$
   - (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ).
   - Update Augmented Lagrangian parameters.
   - Set $j \leftarrow j + 1$.
4. Set $x_{\text{min}} \leftarrow x^{j+1}$ as the best approximation found.

Lino Costa (UMinho - PT)

Genetic Pattern Search Augmented Lagrangian
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi^j(x)$, are solved by GA and HJ.

**GAPSAL**

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   
   - $y^j \leftarrow GA(x^j)$
     (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA).
   - $x^{j+1} \leftarrow HJ(y^j)$
     (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ).
   - Update Augmented Lagrangian parameters.
   - Set $j \leftarrow j + 1$.

4. Set $x_{\text{min}} \leftarrow x^{j+1}$ as the best approximation found.
Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm

Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi_j(x)$, are solved by GA and HJ.

**GAPSAL**

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   - $y^j \leftarrow GA(x^j)$
     (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA).
   - $x^{j+1} \leftarrow HJ(y^j)$
     (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ).
   - Update Augmented Lagrangian parameters.
   - Set $j \leftarrow j + 1$.
4. Set $x_{\text{min}} \leftarrow x^{j+1}$ as the best approximation found.
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, \( \Phi^j(x) \), are solved by GA and HJ.

**GAPSAL**

1. Randomly initialize an initial point \( x^0 \).
2. Set \( j \leftarrow 0 \) as the iteration counter.
3. While the stopping criteria is not met do
   - \( y^j \leftarrow GA(x^j) \)
     (the point \( x^j \) is introduced in the initial population and the remaining points are randomly generated; \( y^j \) is the best approximation found by GA).
   - \( x^{j+1} \leftarrow HJ(y^j) \)
     (HJ starts the search from the best point found by GA; \( x^{j+1} \) is the best approximation found by HJ).
   - Update Augmented Lagrangian parameters.
   - Set \( j \leftarrow j + 1 \).
4. Set \( x_{\text{min}} \leftarrow x^{j+1} \) as the best approximation found.
Genetic Pattern Search Augmented Lagrangian (GAPSAL)

Augmented Lagrangian subproblems, $\Phi^j(x)$, are solved by GA and HJ.

GAPSAL

1. Randomly initialize an initial point $x^0$.
2. Set $j \leftarrow 0$ as the iteration counter.
3. While the stopping criteria is not met do
   - $y^j \leftarrow GA(x^j)$
     (the point $x^j$ is introduced in the initial population and the remaining points are randomly generated; $y^j$ is the best approximation found by GA).
   - $x^{j+1} \leftarrow HJ(y^j)$
     (HJ starts the search from the best point found by GA; $x^{j+1}$ is the best approximation found by HJ).
   - Update Augmented Lagrangian parameters.
   - Set $j \leftarrow j + 1$.
4. Set $x_{\min} \leftarrow x^{j+1}$ as the best approximation found.
Outline

1 Motivation

2 Augmented Lagrangian Technique

3 Genetic Algorithm (GA)

4 Hooke and Jeeves (HJ) Pattern Search

5 Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm

6 Numerical results

7 Conclusions and Future Work
### Thirteen benchmark problems (coded in AMPL) were considered:

<table>
<thead>
<tr>
<th>Prob</th>
<th>Type of $f(x)$</th>
<th>$n$</th>
<th>$p$</th>
<th>$m$</th>
<th>$n_{act}$</th>
<th>$f_{global}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>g01</td>
<td>quadratic</td>
<td>13</td>
<td>9</td>
<td>0</td>
<td>6</td>
<td>-15.000000</td>
</tr>
<tr>
<td>g02</td>
<td>nonlinear</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.803619</td>
</tr>
<tr>
<td>g03</td>
<td>polynomial</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1.000000</td>
</tr>
<tr>
<td>g04</td>
<td>quadratic</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>-30665.54</td>
</tr>
<tr>
<td>g05</td>
<td>cubic</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5126.4981</td>
</tr>
<tr>
<td>g06</td>
<td>cubic</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>-6961.814</td>
</tr>
<tr>
<td>g07</td>
<td>quadratic</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>6</td>
<td>24.30621</td>
</tr>
<tr>
<td>g08</td>
<td>nonlinear</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0.095825</td>
</tr>
<tr>
<td>g09</td>
<td>polynomial</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>680.6301</td>
</tr>
<tr>
<td>g10</td>
<td>linear</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>7049.2480</td>
</tr>
<tr>
<td>g11</td>
<td>quadratic</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.750000</td>
</tr>
<tr>
<td>g12</td>
<td>quadratic</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.000000</td>
</tr>
<tr>
<td>g13</td>
<td>nonlinear</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0.053945</td>
</tr>
</tbody>
</table>
Since GAPSAL is a stochastic algorithm, we are reporting values for 25 runs:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>g01</td>
<td>-15.00000</td>
<td>-15.00000</td>
<td>-14.99994</td>
<td>-14.99988</td>
<td>-14.99995</td>
<td>0.00003</td>
<td>224237</td>
</tr>
<tr>
<td>g02</td>
<td>0.803619</td>
<td>0.624221</td>
<td>0.545225</td>
<td>0.487877</td>
<td>0.552127</td>
<td>0.02695</td>
<td>278821</td>
</tr>
<tr>
<td>g03</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>0.00000</td>
<td>85991</td>
</tr>
<tr>
<td>g04</td>
<td>-30665.54</td>
<td>-30665.54</td>
<td>-30665.54</td>
<td>-30665.54</td>
<td>-30665.54</td>
<td>0.00000</td>
<td>81089</td>
</tr>
<tr>
<td>g05</td>
<td>5126.498</td>
<td>5126.498</td>
<td>5126.498</td>
<td>5126.498</td>
<td>5126.498</td>
<td>0.00000</td>
<td>81415</td>
</tr>
<tr>
<td>g06</td>
<td>-6961.814</td>
<td>-6961.814</td>
<td>-6961.814</td>
<td>-6961.814</td>
<td>-6961.814</td>
<td>0.00000</td>
<td>23687</td>
</tr>
<tr>
<td>g07</td>
<td>24.30621</td>
<td>24.30621</td>
<td>24.30621</td>
<td>24.30621</td>
<td>24.30621</td>
<td>0.00000</td>
<td>146882</td>
</tr>
<tr>
<td>g08</td>
<td>0.095825</td>
<td>0.095825</td>
<td>0.095825</td>
<td>0.095825</td>
<td>0.095825</td>
<td>0.00000</td>
<td>15050</td>
</tr>
<tr>
<td>g09</td>
<td>680.6301</td>
<td>680.6301</td>
<td>680.6301</td>
<td>680.6301</td>
<td>680.6301</td>
<td>0.00000</td>
<td>109093</td>
</tr>
<tr>
<td>g10</td>
<td>7049.2480</td>
<td>7049.2480</td>
<td>7049.2480</td>
<td>7049.2480</td>
<td>7049.2480</td>
<td>0.00000</td>
<td>111111</td>
</tr>
<tr>
<td>g11</td>
<td>0.750000</td>
<td>0.750000</td>
<td>0.750000</td>
<td>0.750000</td>
<td>0.750000</td>
<td>0.00000</td>
<td>1369</td>
</tr>
<tr>
<td>g12</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>0.00000</td>
<td>693199</td>
</tr>
<tr>
<td>g13</td>
<td>0.053945</td>
<td>0.053950</td>
<td>0.438851</td>
<td>0.438851</td>
<td>0.254099</td>
<td>0.19626</td>
<td>75993</td>
</tr>
</tbody>
</table>
Outline

1. Motivation
2. Augmented Lagrangian Technique
3. Genetic Algorithm (GA)
4. Hooke and Jeeves (HJ) Pattern Search
5. Hybrid Genetic Pattern Search Augmented Lagrangian Algorithm
6. Numerical results
7. Conclusions and Future Work
Conclusions and Future Work

- We proposed a hybrid algorithm for constrained global optimization that combines:
  - Augmented Lagrangian Method;
  - Genetic Algorithm;
  - Hooke and Jeeves Pattern Search.
- Numerical results for a set of test problems seems to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;
- In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:
- perform comparisons with other stochastic approaches and solve other benchmark problems;
- improve the integration of GA and HJ (e.g., hill climbing strategies...).
Conclusions and Future Work

- We proposed a hybrid algorithm for constrained global optimization that combines:
  - Augmented Lagrangian Method;
  - Genetic Algorithm;
  - Hooke and Jeeves Pattern Search.

- Numerical results for a set of test problems seems to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

- In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- perform comparisons with other stochastic approaches and solve other benchmark problems;
- improve the integration of GA and HJ (e.g., hill climbing strategies...).
We proposed a hybrid algorithm for constrained global optimization that combines:

- Augmented Lagrangian Method;
- Genetic Algorithm;
- Hooke and Jeeves Pattern Search.

Numerical results for a set of test problems seem to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- perform comparisons with other stochastic approaches and solve other benchmark problems;
- improve the integration of GA and HJ (e.g., hill climbing strategies...).
Conclusions and Future Work

- We proposed a hybrid algorithm for constrained global optimization that combines:
  - Augmented Lagrangian Method;
  - Genetic Algorithm;
  - Hooke and Jeeves Pattern Search.

- Numerical results for a set of test problems seem to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

- In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- Perform comparisons with other stochastic approaches and solve other benchmark problems;
- Improve the integration of GA and HJ (e.g., hill climbing strategies...).
Conclusions and Future Work

We proposed a hybrid algorithm for constrained global optimization that combines:

- Augmented Lagrangian Method;
- Genetic Algorithm;
- Hooke and Jeeves Pattern Search.

Numerical results for a set of test problems seem to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- perform comparisons with other stochastic approaches and solve other benchmark problems;
- improve the integration of GA and HJ (e.g., hill climbing strategies...).
We proposed a hybrid algorithm for constrained global optimization that combines:

- Augmented Lagrangian Method;
- Genetic Algorithm;
- Hooke and Jeeves Pattern Search.

Numerical results for a set of test problems seem to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- perform comparisons with other stochastic approaches and solve other benchmark problems;
- improve the integration of GA and HJ (e.g., hill climbing strategies...).
Conclusions and Future Work

- We proposed a hybrid algorithm for constrained global optimization that combines:
  - Augmented Lagrangian Method;
  - Genetic Algorithm;
  - Hooke and Jeeves Pattern Search.

- Numerical results for a set of test problems seems to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

- In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:
- perform comparisons with other stochastic approaches and solve other benchmark problems;
- improve the integration of GA and HJ (e.g., hill-climbing strategies).
Conclusions and Future Work

- We proposed a hybrid algorithm for constrained global optimization that combines:
  - Augmented Lagrangian Method;
  - Genetic Algorithm;
  - Hooke and Jeeves Pattern Search.

- Numerical results for a set of test problems seem to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

- In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- perform comparisons with other stochastic approaches and solve other benchmark problems;

- improve the integration of GA and HJ (e.g., hill climbing strategies...).
Conclusions and Future Work

- We proposed a hybrid algorithm for constrained global optimization that combines:
  - Augmented Lagrangian Method;
  - Genetic Algorithm;
  - Hooke and Jeeves Pattern Search.

- Numerical results for a set of test problems seem to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

- In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- perform comparisons with other stochastic approaches and solve other benchmark problems;

- improve the integration of GA and HJ (e.g., hill climbing strategies...).
Conclusions and Future Work

- We proposed a hybrid algorithm for constrained global optimization that combines:
  - Augmented Lagrangian Method;
  - Genetic Algorithm;
  - Hooke and Jeeves Pattern Search.

- Numerical results for a set of test problems seem to show that hybridization provides a more effective tradeoff between exploitation and exploration of the search space;

- In general, GAPSAL exhibited a good performance in terms of accuracy and precision of the optimal approximations obtained.

We intend to:

- perform comparisons with other stochastic approaches and solve other benchmark problems;
- improve the integration of GA and HJ (e.g., hill climbing strategies...).
email:  lac@dps.uminho.pt
Web    http://pessoais.dps.uminho.pt/lac

email:  iapinho@dps.uminho.pt
Web    http://www.norg.uminho.pt/iapinho

email:  emgpf@dps.uminho.pt
Web    http://www.norg.uminho.pt/emgpf

Supported by
FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECHNOLOGIA E ENSINO SUPERIOR Portugal