Multiple Start Branch and Prune Filtering Algorithm for Nonconvex Optimization

Rangaprasad Arun Srivatsan[1] and Howie Choset[1]

Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA,
(rarunsrivatsan@,choset@cs.) cmu.edu

Abstract. Automatic control systems, electronic circuit design, image registration, SLAM and several other engineering problems all require nonconvex optimization. Many approaches have been developed to carry out such nonconvex optimization, but they suffer drawbacks including large computation time, require tuning of multiple unintuitive parameters and are unable to find multiple local/global minima. In this work we introduce multiple start branch and prune filtering algorithm (MSBP), a Kalman filtering-based method for solving nonconvex optimization problems. MSBP starts off with a number of initial state estimates, which are branched and pruned based on the state uncertainty and innovation respectively. We show that compared to popular methods used for solving nonconvex optimization problems, MSBP has fewer parameters to tune, making it easier to use. Through a case study of point set registration, we demonstrate the efficiency of MSBP in estimating multiple global minima, and show that MSBP is robust to initial estimation error in the presence of noise and incomplete data. The results are compared to other popular methods for nonconvex optimization using standard datasets. Overall MSBP offers a better success rate at finding the optimal solution with less computation time.

1 Introduction

In various engineering applications such as automatic control systems, signal processing, mechanical systems design, image registration, etc., we encounter problems that require optimization of some objective function. While many efficient algorithms have been developed for convex optimization, dealing with nonconvex optimization remains an open question [17]. In this work, we introduce a new method for nonconvex optimization, called *multiple start branch and prune filtering algorithm* (MSBP). Compared to popular methods, branch and bound [12], simulated annealing [2], genetic algorithms [5], etc., MSBP only has a few parameters to tune and can provide fast online estimates of the optimal solutions.

We believe that Kalman filter-based methods for nonconvex optimization [28] suffer less from issues surrounding computational efficiency and parameter tuning. Multi-hypothesis filtering [20] and the heuristic Kalman algorithm (HKA) [28, 29] are two popular choices for filtering based methods for nonconvex optimization. Both these methods, as well as MSBP, fall under the category of population based stochastic optimization techniques. MSBP was developed for nonconvex optimization problems where the objective function is available in an analytical form and yet is expensive to evaluate (for example the case of point registration).

Unlike the HKA which starts with one initial state estimate, MSBP starts with multiple such estimates. These are further branched, updated and then pruned to explore the search space efficiently while avoiding premature convergence to a local minimum. A major advantage of MSBP over other methods is the high success rate of estimating all the minima in problems with multiple local/global minima. The MSBP requires tuning of only three intuitive parameters, which makes it easy for a non-expert to use the method.

In this work we evaluate and compare the efficiency of MSBP to other methods on the Griewank function, which is a standard test for nonconvex optimization methods. We also test MSBP on point set registration. This application is specifically chosen to test our algorithm because of its analytical and yet expensive function evaluation which offers practical challenges to most of the existing algorithms for nonconvex optimization. MSBP is tested in the presence of high initial error, multiple global minima, noisy data and incomplete data. In all these cases, MSBP accurately estimates the global minima with a high success rate over multiple runs of the algorithm.

2 Background

In a general setting, an optimization problem consists of finding input variables within a valid domain that minimize a function of those variables. An optimization problem can be represented as

$$\begin{aligned} \text{minimize} \quad & h(\boldsymbol{x}), & \boldsymbol{x} \in \boldsymbol{R}^{n_{\boldsymbol{x}}} \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, & i = 1, \ldots, n_c \\ & e_j(\boldsymbol{x}) = 0, & j = 1, \ldots, n_e. \end{aligned} \tag{1}$$

In Eq. 1, $\boldsymbol{x}$ is the $n_{\boldsymbol{x}}$ dimensional input variable, also known as the optimization variable, h is the objective function to be minimized, $g_i(\boldsymbol{x})$ and $e_j(\boldsymbol{x})$ are the inequality and equality constraints respectively and n_c and n_e are the number of inequality and equality constraints respectively.

2.1 Nonconvex optimization problems

We often encounter optimization problems that have a number of locally optimal solutions which are optimal only within a small neighborhood but do not correspond to the globally optimal solution that minimizes the function in the function domain. Such problems are termed “nonconvex” optimization problems, in contrast to “convex” optimization problems where any local minimum is also a global minimum. Nonconvex optimization problems are in general non-trivial to solve because it is difficult to guarantee that the solution returned by the optimizer is global rather than local.

For these problems, a standard approach is to use convex optimizers that employ different randomization techniques to choose multiple initial starts [22]. The drawback of this approach is that for problems with a large number of local minima solutions, a lot of computational effort may be needed to find the global optimum [17]. Branch and bound methods are also commonly used, but the curse of dimensionality leaves them ineffective in cases with many optimization variables [12].

2.2 Heuristic methods for nonconvex optimization problems

Several heuristic methods have been developed to estimate global minima in nonconvex optimization problems such as simulated annealing (SA) [2], particle swarm optimization (PSO) [18], genetic algorithms (GA) [5] and more recently recursive decomposition (RD) [3]. SA is widely considered as versatile and easy to implement, but there are two major drawbacks: 1) there are multiple unintuitive parameters that require tuning, and the results are known to be sensitive to the choice of these parameters [7]; 2) the computation time is generally high for most practical applications [19]. PSO and GA are both categorized as population-based random-search methods. PSO is more sensitive than GA to the choice of parameters, and is known to prematurely converge unless the parameters are tuned well. Also, GA is known to be computationally intractable for many high dimensional problems [25]. In contrast, RD decomposes the objective function into approximately independent sub-functions, and then optimizes the simpler sub-functions using gradient based techniques. The drawback of such a method is that not all functions can be decomposed into sub-functions, in which case RD would perform similarly to a gradient descent with multiple starts.

2.3 Filtering-based methods for nonconvex optimization problems

Due to their ease of use and small number of tuning parameters, Kalman filter-based methods have also been used in optimization [4, 24, 29]. Typically such methods adapt a Kalman filter to have a static process model with the state vector comprised of the optimization variables $\boldsymbol{x}$ and an initial state uncertainty $\boldsymbol{\Sigma}$ spanning the domain of the search space. The measurement model is taken to be an evaluation of the objective function. The measurement is chosen to be the value of the minimum that we want the objective function to attain. By definition, with each iteration of the Kalman filter, the state vector is updated such that the difference between the measurement and the measurement model is decreased [10], thus ensuring that the objective function is minimized. The corresponding covariance also decreases as the number of iterations increases. When the mean of the state stops changing over iterations, or when the uncertainty decreases below a set threshold, we consider the state to be the optimal estimate.

A Kalman filter can provide the optimal estimate of $\boldsymbol{x}_{k+1|k+1}$ [1] such that

$$\boldsymbol{x}_{k+1|k+1} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\ prob(\boldsymbol{x}|\boldsymbol{z}_{k+1}, \boldsymbol{x}_{k+1|k}), \tag{2}$$

where $\boldsymbol{z}_{k+1}$ is the current measurement and $\boldsymbol{x}_{k+1|k}$ is the mean of the predicted state estimate. The solution to Eq. 2 is

$$\begin{aligned}\boldsymbol{x}_{k+1|k+1} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\ & (\boldsymbol{x} - \boldsymbol{x}_{k+1|k})^T \boldsymbol{\Sigma}_{k+1|k}^{-1} (\boldsymbol{x} - \boldsymbol{x}_{k+1|k}) + \\ & (\boldsymbol{z}_{k+1} - h(\boldsymbol{x}))^T \boldsymbol{R}^{-1} (\boldsymbol{z}_{k+1} - h(\boldsymbol{x})),\end{aligned} \tag{3}$$

where $\boldsymbol{\Sigma}_{k+1|k}$ is the uncertainty of the predicted state, and h is the unconstrained objective function as defined in Eq. 1.

Let h_{min} be the smallest value that h can attain, which is attained at $\boldsymbol{x} = \boldsymbol{x}_{min}$. Since there is uncertainty associated with $\boldsymbol{x}$, we have $h_{min} = h(\boldsymbol{x}_{min}) = h(\boldsymbol{x}) + \boldsymbol{v}$, where $\boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{R}(\boldsymbol{x}))$ is state dependent measurement noise drawn from a zero mean distribution with covariance $\boldsymbol{R}$. For an optimization problem as shown in Eq. 1, we choose the following measurement model

$$\boldsymbol{z}_k = h(\boldsymbol{x}) + \boldsymbol{v}_k(\boldsymbol{x}).$$

We set the measurement $\boldsymbol{z}_k = h_{min}$, as the state $\boldsymbol{x}$ will then be updated such that the value of h is close to h_{min}. If h_{min} is not known *a priori*, $\boldsymbol{z}_{k+1}$ can be set to an arbitrarily small value. The uncertainty $\boldsymbol{R}(\boldsymbol{x})$ can be computed analytically as shown in [9, pp. 90–91]. The resulting filter would provide an optimal estimate of $\boldsymbol{x}$ as long as h is linear [10]. The following are the Kalman filter equations modified for an optimization problem:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{K}_{k+1}(h_{min} - h(\boldsymbol{x}_k)), \tag{4}$$

$$\boldsymbol{\Sigma}_{k+1} = \boldsymbol{\Sigma}_k - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1}\boldsymbol{\Sigma}_k, \tag{5}$$

where the Kalman gain $\boldsymbol{K}_{k+1} = \boldsymbol{\Sigma}_k \boldsymbol{H}_{k+1}^T (\boldsymbol{H}_{k+1}\boldsymbol{\Sigma}_k \boldsymbol{H}_{k+1}^T + \boldsymbol{R})^{-1}$ and $h(\boldsymbol{x}_k) = \boldsymbol{H}_k \boldsymbol{x}_k$. $\boldsymbol{H}$ is the Jacobian of the objective function $h(\boldsymbol{x})$.If h is nonlinear, variants such as the extended Kalman filter (EKF), unscented Kalman filter (UKF), etc. can be used. In the presence of constraint functions that must be satisfied as shown in Eq. 1, equality or inequality constrained Kalman filtering techniques can be applied [4,26].

In general, the Kalman filter only estimates the local minimum. A popular approach for nonconvex optimization problems is multi-start or multi-hypothesis filtering as shown in Fig. 1(a) [20]. Multiple filters each having a different randomly chosen initial start, are run in parallel, and after each iteration the estimate with the maximum likelihood is chosen as the current best estimate. Such an implementation has a good chance of finding global minima but at the expense of increased computation time.

Particle filters have also been adapted as a smart alternative to multi-hypothesis filtering [15]. The resampling step in a particle filter ensures that states with low

[1] $\boldsymbol{v}_{a|b}$ is the estimate of $\boldsymbol{v}$ at the a^{th} iteration given measurements upto b iterations.

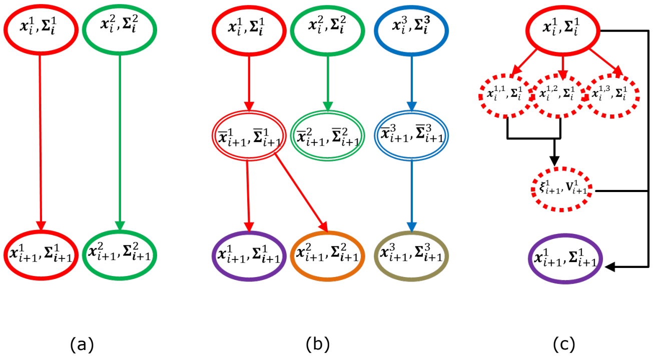
x_i^1, Σ_i^1
x_i^2, Σ_i^2
$x_{i+1}^1, \Sigma_{i+1}^1$
$x_{i+1}^2, \Sigma_{i+1}^2$
(a)
x_i^1, Σ_i^1
x_i^2, Σ_i^2
x_i^3, Σ_i^3
$\bar{x}_{i+1}^1, \bar{\Sigma}_{i+1}^1$
$\bar{x}_{i+1}^2, \bar{\Sigma}_{i+1}^2$
$\bar{x}_{i+1}^3, \bar{\Sigma}_{i+1}^3$
$x_{i+1}^1, \Sigma_{i+1}^1$
$x_{i+1}^2, \Sigma_{i+1}^2$
$x_{i+1}^3, \Sigma_{i+1}^3$
(b)
x_i^1, Σ_i^1
$x_i^{1,1}, \Sigma_i^1$
$x_i^{1,2}, \Sigma_i^1$
$x_i^{1,3}, \Sigma_i^1$
ξ_{i+1}^1, V_{i+1}^1
$x_{i+1}^1, \Sigma_{i+1}^1$
(c)

Fig. 1. (a) Steps involved in one iteration of a multi-hypothesis filter with 2 initial start states. After each iteration the state with maximum likelihood estimate is chosen as the best current estimate. (b) Steps involved in a particle filter with 3 particles. After updating the particles based on the measurement, resampling is performed to remove particles with low weights. (c) Steps involved in one iteration of the heuristic Kalman algorithm. In this example, the parent's state is divided into 3 child states. The weighted sum of 2 child states with the lowest objective value is used to obtain the pseudo measurement $\boldsymbol{\xi}_{i+1}$.

weights are pruned while the others are retained (see Fig. 1(b)). Particle filters and multi-hypothesis filters both suffer from the curse of dimentionality. When estimating high dimensional parameters (> 4), a large number of particles are needed to span the search space to find the global optimum, which can be computationally expensive especially if the function evaluation is not cheap.

The heuristic Kalman algorithm (HKA), introduced by Toscana et al. [28], is a combination of Kalman filtering and population-based random-search methods (see Fig. 1(c)). Starting with a parent state, HKA spans child states and evaluates the function at the child states. A pseudo measurement and its uncertainty ($\boldsymbol{\xi}_{i+1}, \boldsymbol{V}_{i+1}$) are then obtained from the n best states with the smallest function value, and the state ($\boldsymbol{x}_i, \boldsymbol{\Sigma}_i$) is updated using the pseudo measurement. Even though the parent state is divided into a number of child states, in each iteration of the algorithm only a single state, is updated. Such an approach has been shown to be suitable in situations where the function can only be evaluated using experimental simulations and not analytically. For such problems, HKA is a good optimization tool with very few parameters to tune and a good success rate of finding global minimum [29]. However, when an analytical form of the objective function is available, other methods perform much better than HKA.

In this work, we introduce the multiple start branch and prune filtering algorithm (MSBP), which is similar to HKA in that at every iteration the parent state is divided into several child states which are then updated using the current measurement. However, instead of using a weighted mean of the n selected child states to estimate a single next state, we update the m child states based on the measurements and select the n best updated states as parent states for the next iteration and repeat the process until convergence. MSBP also incorporates a procedure to prune the parent states if they are within a threshold of each other. This prevents duplication of computation on similar states and encourages exploration. Such an approach allows us to find multiple global/local minima solutions.

3 Modeling

The basic framework of the MSBP is shown in Fig. 2. The various steps involved in the MSBP implementation are as follows:

1. The algorithm is initialized with n initial parent states $(\boldsymbol{x}_k^i, \boldsymbol{\Sigma}_k^i)$, $i = 1, 2, \ldots, n$, where k denotes the iteration index (see Section 3.1 for information on how to choose the initial states).
2. Each parent state is divided into m child states $(\boldsymbol{x}_k^{i,j}, \boldsymbol{\Sigma}_k^i)$, $(j = 1, 2, \ldots, m)$, by sampling from the distribution $(\boldsymbol{x}_k^i, \boldsymbol{\Sigma}_k^i)$. The parent state is always retained as one of the m child states. The child states that are generated from the parent state can be viewed as perturbations being added to the states to overcome local minima and to encourage exploration.
3. The child states are then updated using Eq. 4 to obtain $(\boldsymbol{x}_{k+1}^{i,j}, \boldsymbol{\Sigma}_{k+1}^{i,j})$.
4. From the $n \times m$ child states, the n states with the lowest innovation, i.e., $\boldsymbol{z}_{k+1} - h(\boldsymbol{x}_{k|k+1})$ from Eq. 3 are chosen as parents for the next iteration.
5. Among the n parent states chosen, if the means of any states come within an ϵ threshold of each other, the state with the lower innovation is retained and the others are pruned (n decreases every time pruning happens).
6. Steps 2-5 are repeated until convergence or up to a fixed number of iterations.

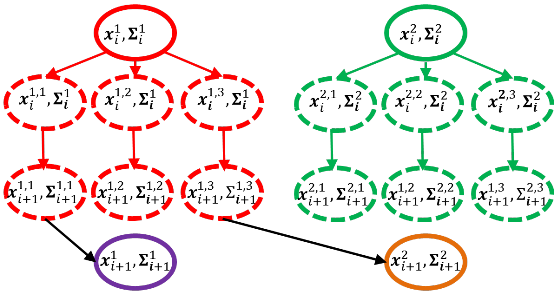
$\boldsymbol{x}_i^1, \boldsymbol{\Sigma}_i^1$
$\boldsymbol{x}_i^2, \boldsymbol{\Sigma}_i^2$
$\boldsymbol{x}_i^{1,1}, \boldsymbol{\Sigma}_i^1$
$\boldsymbol{x}_i^{1,2}, \boldsymbol{\Sigma}_i^1$
$\boldsymbol{x}_i^{1,3}, \boldsymbol{\Sigma}_i^1$
$\boldsymbol{x}_i^{2,1}, \boldsymbol{\Sigma}_i^2$
$\boldsymbol{x}_i^{2,2}, \boldsymbol{\Sigma}_i^2$
$\boldsymbol{x}_i^{2,3}, \boldsymbol{\Sigma}_i^2$
$\boldsymbol{x}_{i+1}^{1,1}, \boldsymbol{\Sigma}_{i+1}^{1,1}$
$\boldsymbol{x}_{i+1}^{1,2}, \boldsymbol{\Sigma}_{i+1}^{1,2}$
$\boldsymbol{x}_{i+1}^{1,3}, \Sigma_{i+1}^{1,3}$
$\boldsymbol{x}_{i+1}^{2,1}, \boldsymbol{\Sigma}_{i+1}^{2,1}$
$\boldsymbol{x}_{i+1}^{1,2}, \boldsymbol{\Sigma}_{i+1}^{2,2}$
$\boldsymbol{x}_{i+1}^{1,3}, \Sigma_{i+1}^{2,3}$
$\boldsymbol{x}_{i+1}^1, \boldsymbol{\Sigma}_{i+1}^1$
$\boldsymbol{x}_{i+1}^2, \boldsymbol{\Sigma}_{i+1}^2$

Fig. 2. Steps involved in one iteration of the MSBP. Parent states are shown in bold ellipses and child states are shown in dashed ellipses. In this example, $n = 2$, $m = 3$.

From Fig. 1(a) it can be noted that multi-hypothesis filtering is a special case of MSBP with $m = 1$ and $\epsilon = 0$. The multi-hypothesis filter requires a large number of initial states to converge onto a global minimum, as a lack of perturbation can result in premature convergence to a local minimum. Also lack of a pruning step in the multi-hypothesis filter often results in duplication of estimates by multiple filters. Particle filters prune states with lower probability during the resampling step and offer an advantage over multi-hypothesis filter. However, particle filters lack the perturbation step and the state update step present in MSBP, which helps in over coming local minima and quick convergence to the optimal solutions. In comparison to other methods, such as GA, SA, PSO, etc., at each iteration in addition to evaluation of the objective function at multiple states, the states themselves are updated by the update model of the MSBP. While this could be viewed as additional computation, the update step allows us to minimize the function faster and quickly identify the minima compared to the other methods. MSBP provides a maximum of n estimates after each iteration as opposed to a single estimate provided by HKA (see Fig. 1(c)). This is a drawback for HKA in problems that have multiple global minima, as HKA would tend to return an estimate that is at a location intermediate to both the minima. Running the HKA multiple times with different initial states can improve the success rate of finding the global minimum, but at the cost of increased computation time.

Thus, the MSBP offers the advantage of reduced computational load and memory storage in addition to a higher success rate of estimating the global minima, for problems with analytical objective functions. The only shortcoming is that when dealing with very high dimensional systems (typically > 20), the update step of the Kalman filter can become expensive as it would involve inverting a high-dimensional matrix.

3.1 Choice of initial state and parameters

In addition to the choice of initial states, there are three parameters that require tuning in the MSBP: n, m and ϵ. This section describes the intuition behind selecting these parameters and the initial states.

Initial state: In most practical problems the domain of the search space for optimization is known. Without loss of generality, the uncertainty of all the initial states is chosen to be equal to each other. The uncertainty is chosen to be a diagonal matrix with each diagonal element set to be equal to σ^2, such that 6σ equals the span of the domain in that dimension. Such a choice for $\boldsymbol{\Sigma}_0^i$ is generally conservative, and restricts the uncertainty in each of the parent state to the search domain. The mean of the states $\boldsymbol{x}_0^i$ are randomly chosen from the valid search domain.

Number of parent states $\boldsymbol{n}$: n can be chosen based on prior knowledge of the number of global minima present in the problem. If that number is not known *a priori*, then a conservative estimate can be made. In practice we observe that choosing a value of n greater than the number of global and local minima present in the search domain improves the success rate of the algorithm. However, increasing the value of n also increases the computation time.

Number of child states $\boldsymbol{m}$: m is the number of child states per parent state. If the estimator is stuck at a local minima, the perturbations help get it out of the local minima. Hence, the greater the value of m, the greater the chances of MSBP capturing the global minima. However a higher m would also mean increased computation time. As result m should be chosen depending on the allowable computation time for the application.

Choice of $\boldsymbol{\epsilon}$: ϵ is the parameter that decides the threshold between the parent states. ϵ helps prevent unnecessary computation and encourages exploration. A large value of ϵ can prune several parent states at once and can result in missing some solutions. $\epsilon = 0$ would not prune any parent state, resulting in unwanted computation in cases where multiple parent states are identical. Depending on the application, ϵ can be chosen to be a fixed number for all iterations or its value can be varied over the iterations.

Section 4.2 describes in more detail how these parameters are tuned for a case study on point set registration.

4 Results

In this section, we first demonstrate the performance of MSBP by testing it on the Griewank function. Following that, we do a case study of point set registration problem.

4.1 Numerical experiment with Griewank function

A number of standard functions are used to test the performance of nonconvex optimization methods [23]. In this work, we choose to test the MSBP on the Griewank function. Fig. 3(a) shows the plot of Griewank function for $x \in [-60,\ 60]$. In the chosen domain, the function is known to have a global minima at $x = 0$ and twenty local minima at ± 6.28, ± 12.56, ± 18.84, ± 25.12, ± 31.45, ± 37.55, ± 43.93, ± 50.3, ± 56.67.

As mentioned in Section 3.1 in order to ensure that 99% of samples fall within the search domain we choose the uncertainty of the initial states, $\Sigma_0 = \sigma_0^2$, such that $6\sigma_0 = 120$. The mean of the initial parent states are sampled from the normal distribution $\mathcal{N}(\mu, \Sigma_0)$, where $\mu = 5$. We choose $\mu = 5$, as it is closer to the local minima at $x = 6.28$ than the global minima at $x = 0$, and would be a more challenging test for the optimization algorithm. For our implementation of MSBP, we use an EKF since the function is non-linear. In addition we choose $n = 21, m = 10, \epsilon = 2$. We run all the algorithms until convergence. The algorithm is set to have converged when the change in the estimate of the minima is $< 10^{-6}$. We observe that the maximum number of iterations required by any algorithm is generally under 20. For the sake of a fair comparison, the values of the parameters for all the algorithms were tuned as per the recommendation in [28] and the best results have been reported.

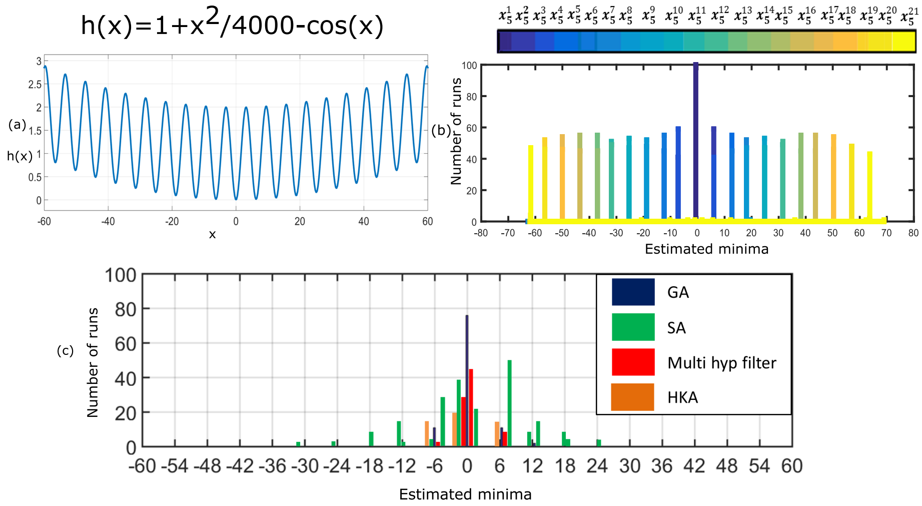

Fig. 3. (a) A plot of the Griewank function. (b) A histogram showing the values estimated by 21 parent states of MSBP over 100 runs. The Y axis of the plot shows the number of runs that estimate a particular state and the X axis shows the estimated value. A histogram of the estimated value over 100 runs is shown for the following algorithms (c) Histogram showing values estimated by Genetic algorithm, Simulated annealing, Multi-hypothesis filter, and HKA.

We repeat the experiment 100 times to observe the performance of the method over multiple runs. Fig. 3(b) shows the histogram of the values estimated by MSBP over 100 runs, all of which converged within five iterations. The global minimum is estimated correctly at $x = 0$ each time, while the local minima solutions are accurately predicted by the remaining twenty parent states. The order in which the other parent states estimate the local minima varies in each run of the algorithm, but they are tracked in all of them.

In comparison, HKA was implemented with initial state $(x_0, \Sigma_0) = (5, 400)$, 20 divisions, 2 best candidates, and a slow down coefficient of 0.7 . Fig. 3(c) shows the histogram of values estimated by HKA over 100 runs. We observe that the algorithm correctly estimates the local minima only 10% of the time. In 8% of the runs, HKA estimates the local minima at $x = 6.28$ and $x = -6.28$, which are closest to the mean of the initial state, $x_0 = 5$. SA also estimates the global minimum only 15% of times and rest of the times it gets stuck at nearby local minima (see Fig. 3(c)).

A multi-hypothesis filter was also implemented, where we choose the same initial states as those for MSBP with $n = 21, m = 1, \epsilon = 0$. Fig. 3(c) shows the histogram of estimated values over 100 repeated runs. More than 50% of the time, the algorithm estimates the global minimum correctly. The rest of the times it estimates values close to the global minimum or one of the two local minima closest to the global minimum similar to the GA (Fig. 3(c)).

4.2 Point set registration: A case study

Point set registration is the process of finding a spatial transformation that aligns the elements of two point sets. Point set registration is frequently encountered in robotic applications, such as computer vision [13], localization and mapping [8], surgical guidance [15], etc. When the correspondence between the points in the two point sets is known, rigid registration can be solved analytically as shown in [6]. However, when point correspondences are unknown, finding the optimal transformation becomes a nonconvex optimization problem with several local minima solutions. Besl et al. came up with the popular iterative closest point (ICP) method that recursively finds correspondences and minimizes the alignment difference between point sets [1]. Over the years several variants of the ICP have been developed [21], and also filtering based solutions have been developed that are better at handling noise in the data and provide online estimates [16].

Most of the point registration methods mentioned above use tools that are not designed for nonconvex optimization and so often converge to local minima. Branch and bound based technique has been developed to avoid this problem [31]. However, this methods has high computation time and is not suitable in real time applications. In this work, we use the MSBP for registration of point sets and demonstrate that it is able to find accurate estimates with low computation times. We perform multiple case studies with different conditions using different standard 3D shape datasets to show the versatility of our algorithm.

We use a dual-quaternion based Kalman filter (DQF) for estimating the registration parameters [27]. The MSBP can also be used with other filtering implementation for registration estimation such as [11,16]. The DQF uses dual-quaternion representation for pose and reposes the originally nonlinear estimation problem as a linear estimation problem and hence Eq. 4 is readily used for estimating the optimal registration parameters (see [27] for more information on the expression for objective function used). In each iteration of the MSBP, closest point correspondence is found between a pair of points (as opposed to finding correspondence for all the points in the case of methods such as ICP). The correspondence found using the parent states is retained for the child states as well. Since the number of different correspondences that can be formed between the two point sets is combinatorial, we expect many local minima solutions. Hence, we choose a large value for n in all the applications below. When the state uncertainty reduces below a desired threshold, we end the estimation process and stop collecting measurements. Thus, compared to batch processing methods such as the ICP (in which we wait for all the measurements to be collected before estimating the optimal registration), the DQF has the advantage of faster computation using fewer point measurements [27].

Large initial transformation error Fig. 4(a) shows the CAD model of a Stanford bunny [30]. The CAD model is geometrically discretized using a triangular mesh with 43318 triangle vertices. We collect 1000 random samples of points from the CAD model and apply a known transformation to those points. We then estimate the applied transformation with the MSBP. The values of

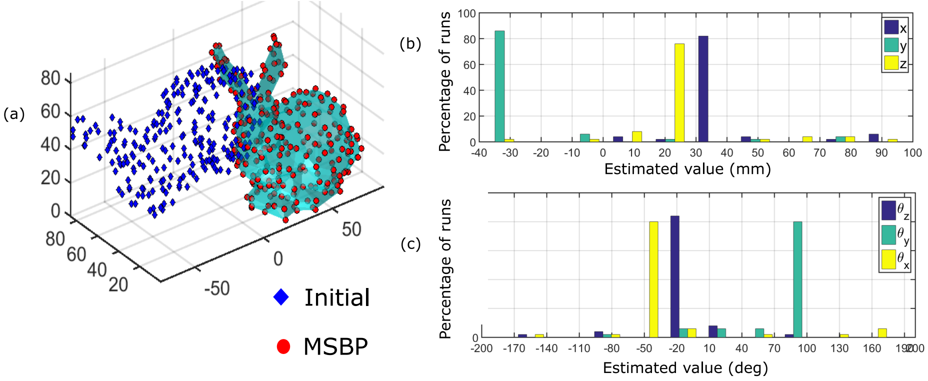

Fig. 4. (a)CAD model of a Stanford bunny. The initial position of 1000 points in shown in blue-diamond markers, the position estimated by MSBP is shown in red-circular markers. (b) Histogram of the estimated translation parameters, (c) histogram of the estimated rotation parameters over 100 runs of the algorithm. In (b) and (c), the Y axis shows the percentage of runs that return a particular value and the X axis shows the estimated value returned by the parent state with the smallest innovation. MSBP has a high success rate of estimating the optimal parameters.

various parameters used are $n = 40, m = 10, \epsilon = 1$. The experiment is repeated 100 times to note the statistical performance of our method. Fig. 4(a) shows the MSBP estimated points lie on top of the CAD model indicating accurate registration. On an average our algorithm converges after using 120 measurements.

Table 1 shows the actual registration parameters and the estimated values. The algorithm is compared with HKA, multi-hypothesis filtering, ICP, SA and GA. The SA and GA implementation we use for the sake of comparison are as described in [14, 25], which are a modified form of the original implementations of SA and GA with internal ICP computations. The authors of [14, 25] show that even though their approaches are expensive per iteration, they result is requirement of fewer iterations over all for convergence, and hence are faster and more accurate at estimating the registration parameters (These observations have been independently verified by us and hence we do not report results for vanilla implementations of SA and GA in this work).

While MSBP and multi-hypothesis filter estimate the registration parameters in a dual-quaternion space, we convert the estimated values into Cartesian coordinates and Euler angles for easy comparison with other methods. The penultimate column and the last column of Table 1 show the RMS error and time taken for various algorithms [2].

For multi-hypothesis filtering, we use the same set of initial states as MSBP. For HKA, ICP and SA we use a 4×4 identity matrix as the initial transformation.

[2] The computational time taken is calculated for script written in MATLAB R2015a software from MathWorks, running on a ThinkPad T450s (20BX0011GE) laptop from Lenovo with 8 GB RAM and intel i7 processor.

Table 1. Registration for large initial transformation error

	x	y	z	θ_x	θ_y	θ_z	RMS	Time
Case1	(mm)	(mm)	(mm)	(deg)	(deg)	(deg)	(mm)	(sec)
Actual	30	-40	15	-55	80	-20	–	–
MSBP	29.89	-39.84	14.67	-58.57	80.59	-23.31	0.48	28
ICP	42.04	-35.22	8.52	17.83	19.21	33.26	35.06	5.82
Multi-hyp	59.79	-20.66	15.26	53.08	-45.58	30.29	18.25	404.62
HKA	-3.97	-17.69	17.45	31.39	31.69	-22.05	53.44	201.97
SA	29.16	-38.34	13.97	-51.75	81.67	-14.49	2.36	353.67
GA	30.08	-39.93	15.05	-54.59	79.92	-19.51	0.08	1051.00

The bounds on the search space are $[-100,\ 100]$ for translation and $[-\pi,\ \pi]$ for rotation around each axis. For HKA we use 40 divisions, 4 best candidates and a slow down coefficient of 0.4. For GA we use an initial population of 100, crossover probability of 0.7 and mutation probability of 0.2. These values for the parameters are tuned as per [29] and the best results are reported.

Fig. 4(a) shows that the displacement between the initial position of the points and their true position on the CAD model, is quite high and ICP does not perform well for such high initial errors. We notice that HKA also does not estimate the transformation accurately, presumably because it gets stuck at a local minimum. MSBP, SA and GA accurately estimate the transformation, however, SA and GA take much more time than MSBP to estimate. Since each function evaluation consists of an iteration of ICP internally, SA and GA both have higher estimation time than MSBP.

The MSBP algorithm is run 100 times and a histogram of the estimated translation and rotation parameters are shown in Fig. 4(b) and Fig. 4(c), respectively, which show that there is a $> 85\%$ chance of MSBP converging to the correct value. In comparison with MSBP, GA has a success rate of 10% and SA has a success rate of 20%. Thus, the MSBP produces accurate and repeatable results with high success rate, despite large errors in initial registration.

Multiple global minima In this example, we consider a snowflake as shown in Fig. 5(a), which has rotational symmetry about an axis passing through its center and perpendicular to its plane. The object is symmetric to its original shape when rotated about this axis by $\pm 60°, \pm 120°$ and $180°$. We sample 100 points from the CAD model of the snowflake and transform those points by a known transformation: $(x, y, \theta_z) = (15\text{mm}, 30\text{mm}, 45°)$. We then use MSBP to estimate the applied transformation. Since the snowflake is 2-dimensional, we restrict ourselves to in-plane registration. We use the following parameter values for MSBP: $n = 100, m = 10, \epsilon = 5$. After 100 iterations, the number of surviving parent states is 16. Fig. 5(b)-(g) show the position of the points after applying a transformation given by the first six parent states as estimated by the MSBP. The first six parent states of the MSBP accurately capture the six global minima

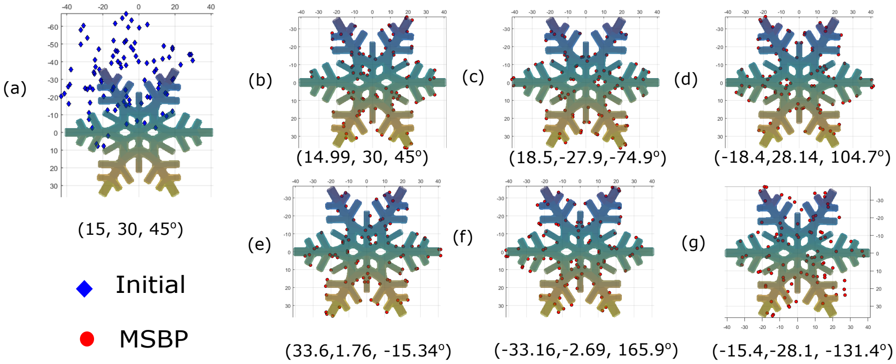

Fig. 5. (a) CAD model of a snowflake.The initial position of 100 points and the position estimated by MSBP are shown in blue-diamond and red-circular markers respectively. The actual transformation between the points and the CAD model is $(15, 30, 45^{\circ})$. (b)-(g) The first six parent states of MSBP. The estimated registration parameters are given below the figure. Note how the rotation angles are $45^{\circ} \pm n \times 60^{\circ}$, $(n = 0, 1, 2)$ due to the 6 way symmetry in the shape of the snowflake. Snowflake CAD model courtesy of Thingiverse CAD model repository

(Note that we limit our search domain to $[-180^{\circ},\ 180^{\circ}]$ and hence there are 6 global minima in the search domain upto the rotational periodicity).

Noise in the input data In order to test the robustness of the registration using MSBP in the presence of noise in one of the point sets, we consider the example of Fertility as shown in Fig. 6. 200 Points are sampled from the CAD model and a Gaussian noise $\mathcal{N}(0, \sigma_n^2)$ is applied to each of the points. The standard deviation σ_n is kept constant for all the points, but is gradually increased from 1 to 20 in increments of 1 over several runs (For reference, the CAD model can be fit in a box of size $300 \times 200 \times 100$ units). Left hand side of Fig. 6(a)-(c) shows that CAD model and the initial position of the points in blue-diamond markers for 3 different values of σ_n. The right hand side of Fig. 6(a)-(c) shows the CAD model and the location of the points after applying the transformation as estimated by MSBP.

Note how the MSBP is able to successfully register the points for all the three cases shown in the figure. Also note how after registration, the points appear to be lying on the CAD model for lower σ_n and appear to be spread out of the CAD model for the case with higher σ_n.

Robustness to incomplete data A number of practical applications that require registration involve partial or incomplete datasets [8]. In order to test the performance of our approach for such applications, we consider an example of Stanford Armadillo man [30] (see Fig. 7). 500 points are sampled from the CAD model. In each run of the algorithm, one point is picked from the point

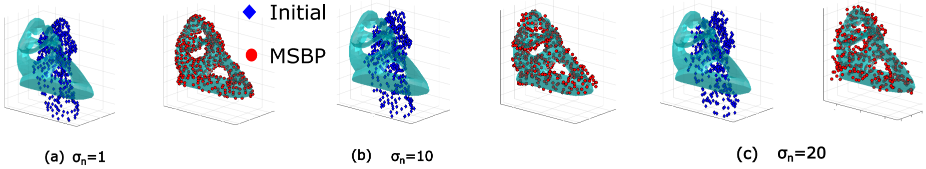

Fig. 6. CAD model of Fertility and 100 points sampled from it and a noise $\mathcal{N}(0, \sigma_n^2)$ is added to the points. (a) the plot for $\sigma_n = 1$ (b) the plot for $\sigma_n = 10$ (c) the plot for $\sigma_n = 20$. CAD model of Fertility courtesy of AIM@SHAPE model repository

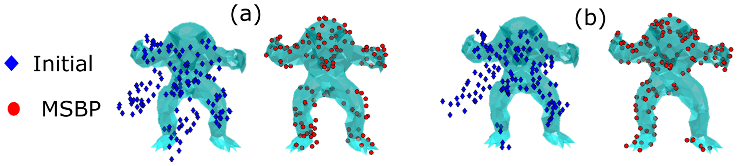

Fig. 7. CAD model of a Stanford Armadillo man [30] and set of initial points sampled from parts of the model. The points are not sampled uniformly from all over the CAD, but have regions of missing information. (a) and (b) show two instances of incomplete data registered accurately to the CAD model using MSBP.

set at random and the selected point along with 250 of its nearest neighbors are removed from the point set. The rest of the points are then used for registration with the original CAD model.

We observe that in spite of the lack of complete point set information, MSBP is able to correctly register the points to the CAD model. Fig. 7 shows two arbitrary runs of the algorithm with different sets of points missing in each. In both the cases, the MSBP correctly registers the points to the CAD model as shown in Fig. 7(a)-(b).

5 Conclusion

In this work, we developed the multiple start branch and prune filtering algorithm (MSBP), a Kalman filter based method for nonconvex optimization. We show that using multiple initial states along with branching, updating and pruning, allows us to efficiently search for the optimal solution(s) in the domain of the search space without prematurely converging to a locally optimal solution. MSBP requires tuning of three parameters, the intuition behind which has been described and empirically verified with several examples. We show that the standard multi-hypothesis filter is a computationally less efficient, special case of the MSBP. With an example of point registration, MSBP is also compared with popular methods for nonconvex optimization and is found to estimate the optimal solutions accurately with a higher success rate especially when: 1) the objective function is available in an analytical form, 2) each function evaluation is expensive, 3) there are multiple global/local minima, and 4) the parameter space is relatively low dimensional (< 20).

Future work will involve an intermediate step to cluster the updated child states instead of using an ϵ threshold. By using an information filter instead of a Kalman filter, the expensive matrix inversion operation step in the state update can be avoided. This would allow us to extend the MSBP for problems involving high dimensional parameter spaces. Validating the effectiveness of MSBP on a variety of nonconvex problems with different functional complexities, different number of parameters, and studying parameter sensitivity, will be a part of our future publication.

Acknowledgments

This work has been funded through the National Robotics Initiative by NSF grant IIS-1426655.

References

1. Besl, P.: A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
2. Brooks, S.P., Morgan, B.J.: Optimization using simulated annealing. The Statistician pp. 241–257 (1995)
3. Friesen, A.L., Domingos, P.: Recursive decomposition for nonconvex optimization. In: Proceedings of IJCAI (2015)
4. Gupta, N., Hauser, R.: Kalman filtering with equality and inequality state constraints. arXiv preprint arXiv:0709.2791 (2007)
5. Holland, J.H.: Outline for a logical theory of adaptive systems. Journal of the ACM 9(3), 297–314 (1962)
6. Horn, B.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A 4, 629–642 (1987)
7. Ingber, L.: Simulated annealing: Practice versus theory. Mathematical and computer modelling 18(11), 29–57 (1993)
8. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM symposium on User interface software and technology. pp. 559–568 (2011)
9. Jazwinski, A.H.: Stochastic processes and filtering theory. Courier Corp. (2007)
10. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82(1), 35–45 (1960)
11. Kang, Z., Chen, J., Wang, B.: Global Registration of Subway Tunnel Point Clouds Using an Augmented Extended Kalman Filter and Central-Axis Constraint. PloS one 10(5) (2015)
12. Lawler, E.L., Wood, D.E.: Branch-and-bound methods: A survey. Operations research 14(4), 699–719 (1966)
13. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence. vol. 81, pp. 674–679 (1981)

14. Luck, J., Little, C., Hoff, W.: Registration of range data using a hybrid simulated annealing and iterative closest point algorithm. In: Proceedings of IEEE International Conference on Robotics and Automation. pp. 3739–3744. IEEE (2000)
15. Ma, B., Ellis, R.E.: Surface-based registration with a particle filter. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 566–573. Springer (2004)
16. Moghari, M.H., Abolmaesumi, P.: Point-based rigid-body registration using an unscented Kalman filter. IEEE Transactions on Medical Imaging 26(12), 1708–1728 (2007)
17. Neumaier, A., Shcherbina, O., Huyer, W., Vinkó, T.: A comparison of complete global optimization solvers. Mathematical programming 103(2), 335–356 (2005)
18. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm intelligence 1(1), 33–57 (2007)
19. Ram, D.J., Sreenivas, T., Subramaniam, K.G.: Parallel simulated annealing algorithms. Journal of parallel and distributed computing 37(2), 207–212 (1996)
20. Reid, D.B.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control 24(6), 843–854 (1979)
21. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Proceedings. Third International Conference on 3-D Digital Imaging and Modeling. pp. 145–152 (2001)
22. Schoen, F.: Stochastic techniques for global optimization: A survey of recent advances. Journal of Global Optimization 1(3), 207–228 (1991)
23. Schoen, F.: A wide class of test functions for global optimization. Journal of Global Optimization 3(2), 133–137 (1993)
24. Schön, T., Gustafsson, F., Hansson, A.: A note on state estimation as a convex optimization problem. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 6, pp. VI–61. IEEE (2003)
25. Seixas, F.L., Ochi, L.S., Conci, A., Saade, D.M.: Image registration using genetic algorithms. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation. pp. 1145–1146. ACM (2008)
26. Simon, D., Chia, T.L.: Kalman filtering with state equality constraints. IEEE transactions on Aerospace and Electronic Systems 38(1), 128–136 (2002)
27. Srivatsan, R.A., Rosen, G.T., Naina, F.M., Choset, H.: Estimating SE(3) elements using a dual-quaternion based linear Kalman filter. in the proceedings of Robotics Science and Systems (2016)
28. Toscano, R., Lyonnet, P.: Heuristic Kalman algorithm for solving optimization problems. IEEE Transactions on Systems, Man, and Cybernetics 39(5), 1231–1244 (2009)
29. Toscano, R., Lyonnet, P.: A Kalman Optimization Approach for Solving Some Industrial Electronics Problems. IEEE Transactions on Industrial Electronics 11(59), 4456–4464 (2012)
30. Turk, G., Levoy, M.: The Stanford 3D Scanning Repository. Stanford University Computer Graphics Laboratory http://graphics.stanford.edu/data/3Dscanrep
31. Yang, J., Li, H., Jia, Y.: Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. In: 2013 IEEE International Conference on Computer Vision (ICCV). pp. 1457–1464 (2013)