

Hybridizing Particle Filters and Population-based Metaheuristics for Dynamic Optimization Problems

Juan José Pantrigo Ángel Sánchez
juanjose.pantrigo@urjc.es angel.sanchez@urjc.es
Dpto. de Informática, Estadística y Telemática
Universidad Rey Juan Carlos
Campus de Móstoles
28933, Madrid, Spain

Abstract

Many real-world optimization problems are dynamic. These problems require from powerful methods to adapt to problem modifications over time. Most applied research on metaheuristics has focused on static (non-changing) optimization problems and these methods often lack from adaptation strategies. Particle filters are sequential Monte Carlo estimation methods which can be applied to Bayesian filtering for nonlinear and non-Gaussian discrete-time dynamic models. In this paper, we propose a general method to hybridize population-based metaheuristics (PBM) and particle filters (PF). The aim of this method is to naturally devise to effective hybrid algorithms to solve dynamic optimization problems by exploiting the benefits of both approaches. Derived algorithms cleverly combine PF and PBM frameworks. As particular examples, two different effective algorithms, named Path Relinking Particle Filter (PRPF) and Scatter Search Particle Filter (SSPF) are respectively derived from the proposed hybridization method. Finally, efficient applications of these instantiated algorithms to different dynamic problems are also presented.

1 Introduction

Dynamic optimization problems appear in many real-world applications [15][3]. Compared to the amount of research undertaken on static optimization, relatively little work has been devoted to dynamic problems [3]. In fact, dynamic problems often lack of well defined optimization functions, standard benchmarks or criteria for comparing solutions [15]. Dynamic optimization problems are characterized by an initial problem definition and a collection of “events” over the time defining some changes on the problem environment [15].

In standard (static) optimization problems, metaheuristics converge near to an optimum, but this solution usually loses its diversity in the intensification stage. Therefore, these methods lack from an adaptation capability regarding to problem domain changes [3]. Restarting the search procedure after changes is the simplest way to deal with dynamic environments. Nevertheless, subsequent events will be processed as unrelated in this approach, thus rejecting useful information and increasing the computation time.

Assuming non-random changes, the new optimum location should be related with the solution of the previous problem definition. Thus, the actual optimization process could use previous knowledge for a more efficient search in the solution space. Population-based metaheuristics, for example, perform this knowledge transfer by including some individuals from the previous population in the new one [3]. Constructive methods allow to adjust solutions using a deconstruction-reconstruction procedure [15].

On the other hand, many dynamic problems require the estimation of the system state that changes over time using a sequence of noisy measurements made on the system [1]. The filtering problem concerns about updating the present state of knowledge and predicting about the future state of the system [5]. In some dynamic problems, previous knowledge about the phenomenon is available, thus allowing to develop models relating unknown quantities with the observations. As a result, the complete solution to the estimation problem is described by a probability density function (pdf) containing all available statistical information. Particle filters are population-based sequential Monte Carlo estimation algorithms where the pdf is approximated by simulated random measures called particles [5]. Each particle represents a weighted solution to the estimation problem.

Dynamic optimization problems use optimization techniques, but also require prediction techniques. The key aspect is how to efficiently use relevant information from pre-

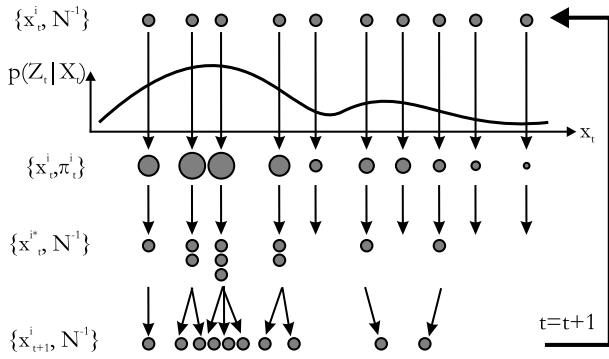


Figure 1. Particle Filter scheme.

vious time steps to find high quality solutions in the subsequent stages. Classical metaheuristics are effective optimization methods, that are oriented to static problems. By the other side, particle filters are specialized in sequential estimation problems, providing prediction strategies. Our work aims to the natural combination of both approaches to devise more efficient hybrid algorithms for dynamic optimization problems.

2 Particle Filters

To make inference about a dynamic system, two different models are required: (i) a measurement model requiring an observation vector Z_t and a system state vector X_t , and (ii) a system model describing the evolution in time of the system state [1][7].

The objective of the Bayesian approach to dynamic state estimation is to construct the posterior pdf representing the state evolution. Usually, an estimate is required at every time step. In the Sequential Bayesian Modelling framework, the posterior pdf is estimated in two stages: prediction and update, respectively.

Particle filters (PF) are a special class of sequential estimation methods in which theoretical distributions on the state space are approximated by simulated random measures called particles [5]. Each particle has a state x_t^i and a weight π_t^i .

Figure 1 shows an outline of the PF approach. PF algorithm starts by setting up an initial population X_0 of N particles using a known pdf. The measurement vector Z_t at time step t , is obtained and particle weights Π_t are computed using a fitness function. These weights are normalized and a new particle set X_t^* is selected. As particles with larger weight values can be selected several times using a Monte Carlo selector, a diffusion stage is applied to avoid the loss of diversity in X_t^* . Finally, the particle set, X_{t+1} is predicted using a motion model. Therefore, PF can be seen as algorithms handling the particles time evolution [5].

3 Population-based Metaheuristics

Metaheuristics approximation algorithms combining basic heuristic methods in high level frameworks aimed at efficiently exploring a search space [2]. Metaheuristics have been successfully applied to optimization problems. Population-based metaheuristics (PBM) [2] are algorithms that work with more than one solution at the same time. These methods efficiently perform search tasks oriented towards obtaining a set of high-quality solutions in the search space. Evolutionary Computation (Genetic and Memetic Algorithms), Scatter Search, Path Relinking or Ant Colony Optimization are some of the most studied population-based metaheuristics [2].

4 Metaheuristic Particle Filter method

The hybrid metaheuristic particle filter (MPF) is a method for devising hybridized algorithms between PBM and PF. It is necessary to difference the stages of *initialization* and *optimization* in the PBM, and the stages of *initialization*, *update* and *prediction* in the PF algorithm. Besides, each individual of the population represents a complete solution to the considered problem.

4.1 Hybridization Method

Hybrid metaheuristics (HM) are a skilled combination of elements belonging to different ones. This hybridization provides a higher flexibility and a more efficient behavior than a single metaheuristic when solving complex real-world problems. Talbi [16] presented a complete taxonomy of hybrid metaheuristics as an attempt to provide classification mechanisms and a common terminology. In our proposal, a population-based metaheuristic is naturally embedded into the general PF scheme, by means of a low-level hybridization, with the aim to improve the solution quality of the considered dynamic problem.

The MPF hybridization method, provides a smart and effective way to combine both PF and PBM, as follows:

1. **Include the selected PF algorithm in *initialization*, *prediction* and *update* stages of the MPF template:** First, a PF algorithm is selected. Then, the set of initialization procedures are identified and set as a *PFinitialize* procedure. In the same way, *PFpredict* and *PFupdate* represent the prediction and updating tasks, respectively. This stages can be easily identified in the PF-based algorithms.
2. **Include the selected PBM in *initialization* and *optimization* stages of the MPF template:** A population-based metaheuristic is selected. Then, the set of initialization and optimization procedures are identified and

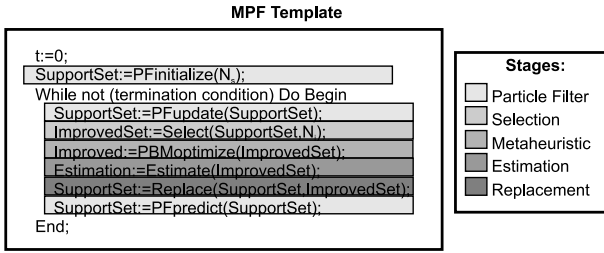


Figure 2. Metaheuristic Particle Filter Template

set as *PBMinitialize* and *PBMupdate*, respectively. These two stages were found in all studied PBM.

3. **Develop a selection procedure:** The selection procedure consists in choosing a subset *ImprovedSet* of N_i solutions from *SupportSet* of N_s solutions. This procedure is mainly dependent on the PBM employed.
4. **Develop an replacement procedure:** Improved solutions from *ImprovedSet* are inserted in *SupportSet* following some criteria. As for the sake of simplicity, N_s must remain constant during all the algorithm, some existing solutions in *SupportSet* are replaced by other solutions with higher fitness values.
5. **Select an estimation procedure:** This procedure is oriented to provide a system state estimate at each time step.
6. **Arrange the different procedures using the MPF template:** The different elements are instantiated to obtain a particular hybrid algorithm using the MPF template shown in Figure 2

4.2 Main Features of Hybrid MPF Algorithms

The resulting algorithms are hybrid methods which can be applied to a wide variety of dynamic optimization problems. They handle a set of solutions in two different stages:

- In the *PF stage*, a solution set, called *SupportSet* of size N_s is propagated and updated to obtain a new one in every time step. This stage is focused on the temporal evolution in time of the best solutions found in previous time steps. The aim when using PF is both tracking multiple hypotheses and using the knowledge about the system dynamics.
- In the *PBM stage*, a new set called *ImprovedSet* is created by selecting a fixed number of N_i solutions

from *SupportSet*. Solutions in *ImprovedSet* are chosen and combined to obtain new better ones.

Hybrid MPF instances lead the search process to a region of the solution space in which it is highly probable to find new better solutions than the previously computed ones. PF stage increases the performance of general PBM in dynamic optimization problems by improving the quality of the diverse initial solution set (*ImprovedSet*).

In order to obtain the initial diverse solution set (*SupportSet*) in time step $t+1$, PF performs two tasks over the *SupportSet* in time step t : 1) select the best solutions and 2) predict new solutions from these best ones. First, a selection procedure chooses solutions with larger weights more likely than those with lower ones. In addition, a diffusion procedure is applied to the selected solutions in the *SupportSet* to avoid the degeneracy problem. This stage provides the required diversity in *SupportSet* to adapt it to severe dynamic changes. Second, the PF performs a prediction procedure over these best solutions to obtain *SupportSet* in time $t+1$. In this way, PF tackles with problem changes in time by predicting the best time evolution of the solutions. As a result, *ImprovedSet* solutions in $t+1$ will be closer to the global optimum than another randomly obtained ones.

Therefore, MFP hybrid algorithms adapt computational load to problem constraints, by reducing the number of required evaluations of the weight function. In this way, solutions in *ImprovedSet* will be selected from a better solution set. This is the main reason why hybrid algorithms reduce the required number of evaluations by the fitness function, and hence the computational load.

PBM and PF are related in such a way that when the PBM improves its results, the PF performance also improves, and vice versa. PF allows a system parameter tuning in order to adjust the quality and the diversity of *SupportSet*, used as diverse solution set by PBM. On the other hand, PBM improves the quality of the particle set allowing a better estimation of the pdf, by including *ImprovedSet* solutions in the *SupportSet*. These facts yield to general and highly configurable algorithms suited to the particular dynamic problem being solved. Table 1 compares MPF, PB and PBM algorithms with respect to the presence or not of different features.

5 Examples of Hybrid PBM Algorithms

Two different hybrid algorithms are presented in this section: Scatter Search Particle Filter [11][12] and Path Relinking Particle Filter [13][14][12]. Both particular algorithms are obtained by direct instantiation from the general MPF method.

Table 1. PF, PBM and MPF main features

	PF	PBM	MPF
Population-based method	✓	✓	✓
Competition mechanisms	✓	✓	✓
Collaboration mechanisms		✓	✓
Adaptive mechanisms	✓		✓
Exploration mechanisms		✓	✓
Multiple hypotheses handling	✓		✓

5.1 Scatter Search Particle Filter

The Scatter Search Particle Filter (SSPF) algorithm integrates both Scatter Search (SS) and Particle Filter (PF) frameworks in two different stages. Figure 3 shows the hybridization of SS and PF algorithms to obtain the SSPF algorithm, using the template of figure 2:

1. The Condensation algorithm is selected as a particular PF algorithm. Then, the set of initialization, prediction and updating tasks procedures are identified, as shown in Figure 3.
2. The SS approach is considered as a PBM. Then, the set of initialization and optimization procedures are identified.
3. SS requires good quality and diverse solutions in the reference set (*ImprovedSet*). Thus, the *SelectWithDiversity* method is used as selection procedure. It consists in the selection of the $N_i/2$ best solutions and the $N_i/2$ most diverse ones from the *ImprovedSet*.
4. Once the SS stage is finished, the N_i “worst” solutions in *SupportSet* are replaced by the N_s *ImprovedSet* solutions
5. The selection of the best solution found in the SS procedure is used as final solution estimate at each time step.
6. The above components are organized to obtain the derived hybrid algorithm according to the MPF template of Figure 2

In the pseudocode of Figure 3, conventional terminology of PF and SS methods is maintained. Algorithmically, SSPF starts with an initial population size of N_s ($N_s \equiv N$ in Figure 3) random solutions drawn from a known pdf. Solution weights are computed using a weighting function. SS stage is later applied to improve the best obtained solutions in *SupportSet* ($X_t \equiv SupportSet$ in Figure 3). A *ImprovedSet* (*ImprovedSet* $\equiv RefSet$ in the SS framework) is created by selecting a subset of N_i

($N_i \equiv b < N_s$) solutions from *SupportSet*. This subset is composed by the $b/2$ best solutions and the $b/2$ most diverse ones of the *SupportSet*. New solutions are generated and evaluated by combining all possible pairs of particles in the *RefSet*. In order to increase the solution fitness, a improvement method is applied to every new solution. Worst solutions in the *RefSet* are replaced by new better ones when they are found. SS stage ends when new generated solutions in *RefSetNew* do not improve the quality of the previous *RefSet*. Once the SS stage is finished, the “worst” solutions in the *SupportSet* X_t are replaced by the *RefSet* solutions. Then, a new population of solutions X_t^* is created by selecting solutions from X_t with a probability according to their weights. Finally, solutions are projected into the next time step by using the prediction stage. The main considered SSPF algorithm parameters are:

- The number of solutions (N_s) in the *SupportSet*. These should be enough to avoid the loss of diversity. The value of N depends on the problem instance complexity.
- The number of solutions (N_i) in the *ImprovedSet*.
- The amplitude A of the diffusion stage. The diffusion stage is applied to avoid the loss of diversity in *SupportSet*. It is performed by computing a random or Gaussian displacement over the particles with maximum amplitude of A .

5.2 Path Relinking Particle Filter

In a similar way, the Path Relinking Particle Filter (PRPF) algorithm is obtained by hybridizing both Path Relinking (PR) and Particle Filter (PF) algorithms, using the MPF method:

1. The Condensation algorithm is selected again as the PF algorithm.
2. PR is considered in this algorithm as PBM. The set of initialization and optimization procedures are identified.
3. PR requires good quality solutions in the reference set (*ImprovedSet*). Thus the *SelectBest* method is used as selection procedure. It consists on the selection of the N_i best solutions from the *ImprovedSet*.
4. Once the PR stage is finished, the N_i “worst” solutions in *SupportSet* are replaced of the *ImprovedSet* solutions (replacement procedure).
5. The best found solutions in the PF procedure are selected as final system solution in each time step (estimation procedure).

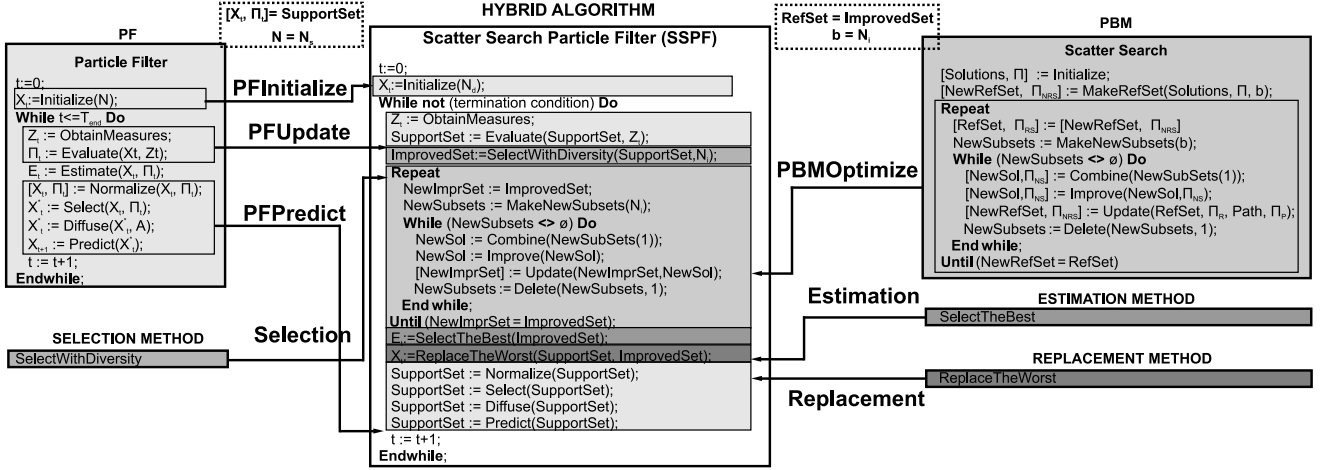


Figure 3. SSPF instantiation from Scatter Search and Particle Filter algorithms.

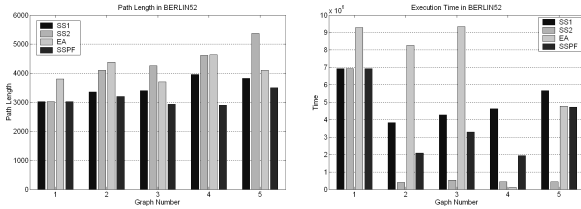


Figure 4. Path length obtained (left) and computational load (right) using two based SS algorithms, GA and SSPF for the benchmark problem *BERLIN52* dynamic adaptation.

- Finally, the above components are organized to obtain the derived hybrid algorithm using the MPF template of Figure 2.

Algorithmically, SS and PR metaheuristics are very similar. The main difference between SS and PR is the combination method to yield new better solutions [2]. The main PRPF algorithm parameters are the same used in the SSPF algorithm.

6 Applications

The proposed hybrid MPF algorithms are suited for many dynamic optimization problems. This section illustrates some of these applications for some different dynamic problems.

6.1 Dynamic Travelling Salesman Problem

The Travelling Salesman Problem (TSP) consists of finding the shortest tour connecting a fixed number of locations (cities), visiting each city exactly once [8]. This problem can be represented by a graph $G = \{V, E, W\}$, where V is a set of vertex representing the cities, E is a set of edges that models the paths connecting cities and W is a symmetric matrix of weights. We suppose that there is an edge connecting every pairs of cities. Each edge weight $w_{i,j} \in W$ represents the distance between two cities $i, j \in V$. The TSP can be described as the problem of finding a Hamiltonian circuit with minimum length in G [17]. The Dynamic TSP (DTSP) is a generalized version of the TSP where G is time-dependent. This problem has several practical applications such as traffic jam analysis [8] or fluctuating a set of active machines [3]. In this work, the considered DTSP approach maintains constant the number of cities, allowing some weight changes among them [8].

In [11] the SSPF algorithm was applied to the DTSP. Experimental results have shown that SSPF appreciably increases the performance of derived SS and EA methods in a challenging dynamic optimization problem, using standard graph benchmarks and without losing quality in the estimation procedure (see Figure 4). This improvement is more significant as the size of the graph problem increases.

6.2 Articulated and Multiple Object Tracking

Most studies in human motion analysis are based on articulated models that properly describe the human body position. When articulated models for human body representation are used, the main problem is the high dimensionality of the state-space and the huge number of computations



Figure 5. Visual model adjustment using frames using: (a) PF and (b) PRPF.

[6]. Moreover, in the PF approach, the number of needed particles grows with the size of the state-space, as demonstrated in [10]. Unfortunately, PF estimation worsens in high-dimensional search spaces [9]. To address this difficulty, several optimized PF algorithms have been proposed [6][10]. In [13][14][12], we presented a model-based PRPF applied to the 2D human pose estimation problem. Experimental results show that PRPF appreciably improves the performance of general and advanced particle filters (see Figure 5). PRPF increases the accuracy of PF for automatically obtaining the marker coordinates with a more reduced particle set in 2D biomechanical analysis.

Human-Computer Interaction (HCI) tendency addresses to non-contact devices, using perceptual and multimodal user interfaces. That means the system enhances the user to interact without any physical contact, using only voice or gesture recognition [4]. The development of a 2D face and hands tracker based on the SSPF algorithm is presented in [12]. The algorithms have been implemented as a Direct-Show transform filter. The SSPF-based system successfully operates on 320x240 live video in real time (28Hz), by reducing the required number of evaluations of the weighting function without affecting the quality of the estimation.

7 Conclusion

The main contribution of this work is the proposal of the new hybrid metaheuristic particle filter (MPF) method. This synergistic template allows to naturally embed a PBM into the PF approach for improving the quality of the particle set. Our aim is the instantiation of new algorithms using functions from different complementary estimation and optimization frameworks to efficiently solve dynamic problems. We have applied our approach for a straightforward derivation of the PRPF and the SSPF algorithms. Experimental results have demonstrated that these algorithms work very well when applied to specific dynamic optimization prob-

blems, in particular, the Dynamic TSP, the Articulated Motion Tracking and Multiple Objects Tracking, respectively. As future work, we propose the use of the MPF method to obtain different hybrid algorithms to solve another practical dynamic optimization problems (in particular, the Dynamic Vehicle Routing Problem).

References

- [1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *IEEE Tr. on Sig. Proc.*, 50(2):174–188, 2002.
- [2] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.
- [3] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer, 2002.
- [4] J. Buades, F. Perales, and J. Varona. Real time segmentation and tracking of face and hands in vr applications. In *LNCS 3179*, pages 259–268, 2004.
- [5] J. Carpenter, P. Clifford, and P. Fearnhead. Building robust simulation-based filters for evolving data sets, 1999.
- [6] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. of the IEEE Conf. on CVPR*, volume 2, pages 126–133, 2000.
- [7] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [8] C. Eyckelhof and M. Snoek. Ant systems for a dynamic tsp: Ants caught in a traffic jam. In *LNCS 2463*, pages 88–99, 2002.
- [9] D. A. Forsyth and J. Ponce, editors. *Computer Vision: A Modern Approach*. Prentice-Hall, 2002.
- [10] J. MacCormick and A. Blake. Partitioned sampling, articulated objects and interface-quality hand tracking. In *Proc. of the 7th ECCV*, volume 2.
- [11] J. J. Pantrigo, A. Duarte, Á. Sánchez, and R. Cabido. Scatter search particle filter to solve the dynamic travelling salesman problem. In *LNCS 3448*, pages 177–189, 2005.
- [12] J. J. Pantrigo, A. S. Montemayor, and R. Cabido. Scatter search particle filter for 2d real-time hands and face tracking. In *LNCS 3617*, pages 953–960, 2005.
- [13] J. J. Pantrigo, Á. Sánchez, K. Gianikellis, and A. Duarte. Path relinking particle filter for human body pose estimation. In *SSPR/SPR LNCS 3138*, pages 653–661, 2004.
- [14] J. J. Pantrigo, Á. Sánchez, K. Gianikellis, and A. S. Montemayor. 2d human tracking by efficient model fitting using a path relinking particle filter. In *LNCS 3179*, pages 202–213, 2004.
- [15] M. Randall. Constructive meta-heuristics for dynamic optimization problems, 2002.
- [16] E. G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8:541–564, 2002.
- [17] H. Zhang-Can, H. Xiao-Lin, and C. Si-Duo. Dynamic traveling salesman problem based on evolutionary computation. In *Proc. 2001 Conf. on Evol. Comp.*, volume 2.