Trajectory Optimization under Changing Conditions through Evolutionary Approach and Black-Box Models with Refining

Karel Macek[1,2], Jiří Rojíček[1], and Vladimír Bičík[1]

[1] Honeywell Labs, V Parku 2326/18, Prague 4 - 148 00, Czech Republic,
{karel.macek, jiri.rojicek, vladimir.bicik}@honeywell.com
[2] Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, Prague 8 - 182 08, Czech Republic

Abstract. This article provides an algorithm that is dedicated to repeated trajectory optimization with a fixed horizon and addresses processes that are difficult to describe by the established laws of physics. Typically, soft-computing methods are used in such cases, i.e. black-box modeling and evolutionary optimization. Both suffer from high dimensions that make the problems complex or even computationally infeasible. We propose a way how to start from very simple problems and - after the simple problems are covered sufficiently - proceed to more complex ones. We provide also a case study related to the dynamic optimization of the HVAC (heating, ventilation, and air conditioning) systems.

Keywords: Empirical function minimization, black-box modeling, simplification, refining, dynamic building control.

1 Introduction

Real world application scheduling and planning suffers from the lack of knowledge of the system structure as well as from the well known curses of dimensionality [1, p. 3-6]. There are already some approaches, addressing these issues. For the modeling part, black-box models can be mentioned: e.g. neural networks [2], Gaussian mixtures [3], Gaussian processes [4], or local regression [5]. All those black-box approaches to modeling reach their limits whenever the number of parameters is high in contrast to the quantity and quality of available data.

Another curse of dimensionality is related to complexity of the optimization problems. In case of black-box modeling, it is not possible to assure that the derived optimization problem will be linear, convex, or without local optima. For such cases, wide class of evolutionary techniques has arisen, including differential evolution [6], particle swarm optimization [7] or covariance matrix adaptation [8]. There are also numerous attempts to optimize those optimization techniques, e.g. [9] or [10]. In case of dynamic problems, techniques of approximate dynamic programming are being improved continuously [1]. Even though those methods

and approaches lead to improved results both in testing suites and practical applications, the essential issue remains: the higher dimensions, the more difficult problem.
This paper provides no new method of black-box modeling or evolutionary optimization. We offer a way how to make complex problems computationally feasible while the user might decide about the simplification. After the system is able to solve simpler problems, the input space might be step-by-step refined, possibly to the original problem.
The paper is organized as follows: We start in Sect. 2 with a very simple case study that motivates the methodology. In Sect. 3, we introduce formal notation that is used in Sect. 4 for the problem formulation. Consequently, the basic approach is provided in Sect. 5 and extended in Sect. 6. Then, the case study in Sect. 7 is used for numerical illustration of the approach. The work is concluded in Sect. 8 where also further work is sketched out.

2 Case Study

We introduce the case study before the formal notation in order to improve the readability of the abstract definitions. Let us consider a building control system with indoor zone temperature T_{za} which stands for the temperature in the room. We will assume a hot season and the upper zone temperature T_{zau}. The control is considered for the whole day, i.e. 24 time instants. Thus we write $T_{za,t}$ and $T_{zau,t}$, where $t = 1, 2 \ldots, 24$. The comfort that have to be assured is defined as $T_{za,t} \leq T_{zau,t}$. Whenever $T_{za,t} > T_{zau,t}$, chillers are started in order to cool the zone down so $T_{za,t} = T_{zau,t}$. There is related power consumption and cost to this operation. Typically, the $T_{zau,t}$ is defined as high as allowed, i.e. $T_{zau,t} = T_{\max,t}$. However the pre-cooling, i.e. choosing $T_{zau,t} < T_{\max,t}$ might be beneficial due to more appropriate ambient profiles of dynamic power prices. This fact motivates the optimization of $T_{zau,t}$, see also [11].
The goal of this case study is to determine $(T_{zau,t})_{t=1}^{24}$ that will minimize the overall costs of the building operation while keeping $T_{zau,t}$ in bounds, i.e.:

$$T_{\min,t} \leq T_{zau,t} \leq T_{\max,t} \tag{1}$$

Note that the case study omits important facts which are considered in other publications, such as internal heat gains, or indoor thermal capacity [12], since we are striving to provide an approach that is able to optimize the system with limited knowledge only.

3 Notation

In this Section, formal notation is introduced and demonstrated in the relationship to the above discussed case study.

Definition 1 (Trajectory). *Let $n_x \in \mathbb{N}$ and $n_t \in \mathbb{N}$. The matrices $X \subset \mathbb{R}^{n_t, n_x}$ are called trajectories, their row indices are called times and column indices inputs. The elements of the trajectories will be denoted as $x_{t,i}$.*

In our case study, we have $n_x = 1$, $n_t = 24$, $x_t \equiv T_{zau,t}$, and X is given by (1).

Definition 2 (Conditions). *Let (Ω, Σ, P) be a probability space called conditions where Ω is a sample space containing all possible outcomes, Σ is set of events where event is subset of Ω, and $P : \Sigma \to [0, 1]$ assigns probabilities to all events.*

The conditions in the considered case study can be both internal (occupancy) and external (weather) factors that influence the operation of the system during next 24 hours.

Definition 3 (Cost function). *Let $c : X \times \Omega \to \mathbb{R}$ be a mapping called cost function.*

The costs in the case study are the costs that the building owner will pay to the utility company. These costs are given by the chiller input power and actual power price.

Definition 4 (Evidence). *Let e be a random vector of size n_e over (Ω, Σ, P), such that exists be called evidence. Let set $E \subset \mathbb{R}^{n_e}$ satisfying $P(E) = 1$ and $\forall \bar{E} \subset \mathbb{R}^{n_e}$ hold $E \subset \bar{E}$.*

Obviously, E is set of all considerable evidences, other have zero probability. The evidence provides information about the future conditions. The evidence in the case study can be embodied as a sequence of weather forecast $\hat{T}_{oa,t}$. The more precise weather forecast, the more valuable the evidence will be.

Definition 5 (Optimal Trajectory). *If a trajectory $x^* \in X$ satisfies for a given evidence e the following condition:*

$$x^* = \arg \min_{x \in X} \mathcal{E} \left[c | e, x \right] \tag{2}$$

we call it optimal for the evidence e.

Definition 6 (Data). *Let us define data as a sequence of triples $D = (\tilde{x}^{(j)}, \tilde{e}^{(j)}, \tilde{c}^{(j)})_{j=1}^{n_j}$ where*

- *matrix $\tilde{x}_j$ is a trajectory used for x_j j-th experiment,*
- *vector $\tilde{e}_j$ is the evidence available for the j-th experiment, and*
- *scalar $\tilde{c}_j$ is the realized cost for the given trajectory and evidence.*

The examples of the optimal trajectory and data for given case study are straightforward: the goal is to set-up the indoor temperature profile minimizing the expected costs. The data records involve the profile used, corresponding weather forecast, and related costs.

4 Problem Formulation

The challenge addressed in this work has two aspects. The first one is that the model of $c|e, x$ is not known exactly (as well as its structure) and we have only limited data set D available. Next issue is related to

the fact that n_t and n_x are relatively large[3]. The first issue makes it difficult to formulate the mathematical problem (2), while the second complicates the search of the optimal solution.
To resolve these issues, we introduce two more concepts:

Definition 7 (Data-Centric Model). *Let $c|e, x, D$ be for each e, x, D a random variable. We denote it as data-centric model.*

Note that this definition involves wide class of models, including those that have been mentioned in the introduction.

Definition 8 (Simplification). *Let $s = (s_x, s_e)$ be a pair of mappings where $s_x : V_x \to X$ where $V_x \subset \mathbb{R}^{n_{sx}}$. Next, $s_e : E \times \mathcal{D} \to V_e$, where $V_e \subset \mathbb{R}^{n_{se}}$ and $\mathcal{D}$ set of all possible data sets. We call s simplification and the set of all considerable simplifications S. A trajectory $x \in X$ satisfies given simplification $s \in S$ iff $\exists v \in V_x : s_x(v) = x$.*

The concept of simplification is intended to reduce the dimension of both black-box modeling and consequent reconstruction of relevant trajectory. Thus, typically $n_{sx} + n_{se} \ll n_e + n_t \cdot n_x$.
In the case study, the simplification is considered as

$$s_x(v_x) = \begin{cases} v_x \text{ for } & t = 1, 2, \dots 8 \\ 24 \text{ for } & t = 9, 10, \dots 22 \\ 30 \text{ otherwise} & \end{cases} \tag{3}$$

where $15 \leq v_x \leq 30$. The s_e could return e.g. the average temperature of the past day (depends on data D) and average value of the forecast for the next day (depends on evidence e). For the visualization purposes we will adopt the average temperature of the past day only.
The quality of the simplification impacts the quality of suboptimal data-centric optimal simplified trajectory, defined as follows, in terms of realized cost:

Definition 9 (Data-Centric Optimal Simplified Trajectory). *Let $s \in S$ be a simplification. Let D be a given data set where all $\tilde{x}_j$ satisfy the simplification s. Let $c|x, e, d$ be a data-centric model for given simplification and given data set. The trajectory $x^* \in X$ satisfying*

$$x^* = s_x(\arg\min_{v \in V_x} \mathcal{E}\left[c|v, s_e(e, D), D_s\right]) \tag{4}$$

where we assume all records in D contain a trajectory satisfying the simplification s and D_s stands for data D transformed by the simplification s_e, s_x^{-1}, is called data-centric optimal simplified trajectory or DCOS trajectory.

DCOS trajectories are the objective of the proposed method. In the case study, we will focus on the optimization of the $T_{zau,t}$ for $t < 9$, i.e. the upper bound between midnight and time when first occupants arrive.

[3] The case study can be extended to optimization of 10 zones with 15 minutes sampling, i.e. $n_t = 24 \cdot 4 = 96$ and $n_x = 10$. Empirical optimization of trajectories with 960 parameters is numerically infeasible.

5 Basic Algorithm

The basic approach can be summarized in the following steps:

1. Define the simplification $s \in S$.
2. Define the data set as empty, i.e. $D = \emptyset$.
3. Set $j = 1$
4. Generate a random trajectory $\tilde{x}^{(1)}$ satisfying s, store the related evidence $\tilde{e}^{(1)}$.
5. Apply the trajectory $\tilde{x}^{(j)}$ to the system and observe the costs $\tilde{c}^{(j)}$.
6. Extend the data set D by new $(\tilde{x}^{(j)}, \tilde{e}^{(j)}, \tilde{c}^{(j)})$.
7. Obtain and store new evidence $\tilde{e}^{(j+1)}$
8. Optimize the new DCOS trajectory $\tilde{x}^{(j+1)}$ for given s, D, and $\tilde{e}_{j+1}$. This involves:
   (a) Simplification of the data using s^+ and s^e to D_s.
   (b) Identification of the model $c|v, s_e(\tilde{e}^{(j)}, D), D_s$
   (c) Solution of $v^* = \arg\min_{v \in V_x} \mathcal{E}\left[c|v, s^e(\tilde{e}^{(j)}, D), D_s\right]$.
   (d) Putting back from the simplified world, i.e. $\tilde{x}^{(j+1)} = s_x(v^*)$
9. Set $j = j + 1$ and go to 4.

Note that during the whole optimization the data are limited to trajectories $\tilde{x_j}$ that satisfy the given simplification s.

6 Refining and More Complex Trajectories

We propose to extend the basic algorithm by application of refining to the used simplification. Let us define us the concept first:

Definition 10 (Refining). *Let $s, \tilde{s} \in S$ be two simplifications. Iff $\forall x \in X$ where s is satisfied also $\tilde{s}$ is satisfied, and there is a surjective mapping $\gamma : \tilde{V}_e \to V_e$, then $\tilde{s}$ is a refining of s.*

The refining might lead to broader set V_x and might lead to better results approaching closer to the minimizer of (2). In the algorithm, the refining can be applied between step 8 and 9. First, it has to be tested whether the refining may be applied. Consequently, the refining has to be chosen. An example of the simplification in the case study would be $\tilde{s} = (\tilde{s_x}, \tilde{s_e})$ where $\tilde{s_e} \equiv s_e$ and $\tilde{V_x} \equiv [15, 30]^2$ and

$$\tilde{s}_x(\tilde{v_x}) = \begin{cases} \tilde{v}_{x,1} & \text{for} \quad t = 1, 2, \ldots 4 \\ \tilde{v}_{x,2} & \text{for} \quad t = 5, 6, \ldots 8 \\ 24 & \text{for} \quad t = 9, 10, \ldots 22 \\ 30 & \text{otherwise} \end{cases} \tag{5}$$

In an analogical way, this refining can be refined again to dimension 4 and again to dimension 8.

6.1 Testing for Refining

Before refining the actual simplification, it has to be assured that the procedure achieves for the given simplification the best possible results. For this purpose we propose the application of the following test. However, other tests can be considered, too. Let $\rho \in \mathbb{R}_+$, let $n_b \in \mathbb{N}$. Let b_j be defined as follows:

$$b_j = \begin{cases} 0 & \text{if} \quad j = 1 \\ b_{j-1} + 1 & \text{if} \quad ||\tilde{x}_j - \hat{x}_j|| < \rho \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\hat{x}_j$ is the DCOS found for $\tilde{e}^{(j)}$, s, and data $D \backslash (\tilde{x}^{(j-1)}, \tilde{e}^{(j-1)}, \tilde{c}^{(j-1)})$. The refining is carried out after $b_j > n_b$. The value of n_b will be discussed in numerical examples in Sect. 7.

6.2 Ways of Refining

The refining is related either to the trajectories X or evidences E. In both cases the refining consists in adding more information into the problem definition. In case of trajectories, the simplification restricts the trajectories to X_1 and the refining uses $X_2 \supset X_1$. In case of evidences, the refining consists in involving additional information. In both cases the refining leads to an increase of the dimension of the black-box model for cost c.

Of course, some refining might lead to no improvements, therefore some testing can be carried out and the simplification might be rejected and another can be tried. However, deeper discussion on this topic is out of the scope of this paper.

7 Numerical Example

The case study has been introduced step-by step in the previous sections. In order to demonstrate the results numerically, we provide basic information about the set-up since detailed description is beyond the scope of this work. At the end of this section, achieved results are discussed.

First, a first-principle simulation model has been adopted for a single zone building with one chiller. We adopted a version of [13] where the thermal capacity of the zone is influenced by ambient air temperature, heat gains from the occupants (people in the zone), internal thermal inertia given by gains and the chiller itself. The parameters have been determined based on experience of modeling of large-scale single zone buildings. The weather profile have been used from a real building, slightly shifted so it had values between 20 and 35°C.

The adopted surrogate black-box model was based on local polynomial regression with Gaussian kernel with covariance matrix $C_{i,i} = 1$ and $C_{i,j} = 0, i \neq j$ and degree 2 (quadratic regression). More details can be found in [14]. The considered model had single output, namely the costs c and several regressors: first was related to the data D while the

others represented v_x. The first regressor was the first PCA [15] component from data containing: (i) weather profile from the last day and (ii) internal temperatures (zone air, building construction). As a procedure for optimization of v_x, the covariance matrix adaptation [8] was selected and the limit of iterations was 200 for each optimization.
We worked with 30 days and carried out 2 experiments with different settings as illustrated on Fig.7. The first experiment with the setting $n_b = 0$ and $\rho = 1$ leads to fast refining and the problem has comparable complexity as if the 8-dimensional problem would be addressed directly. The other experiment with $n_b = 5$ leads to slower refining. It can be observed that better results, i.e. lower costs, are obtained when the dimensionality grows slowly with increasing information in the data.

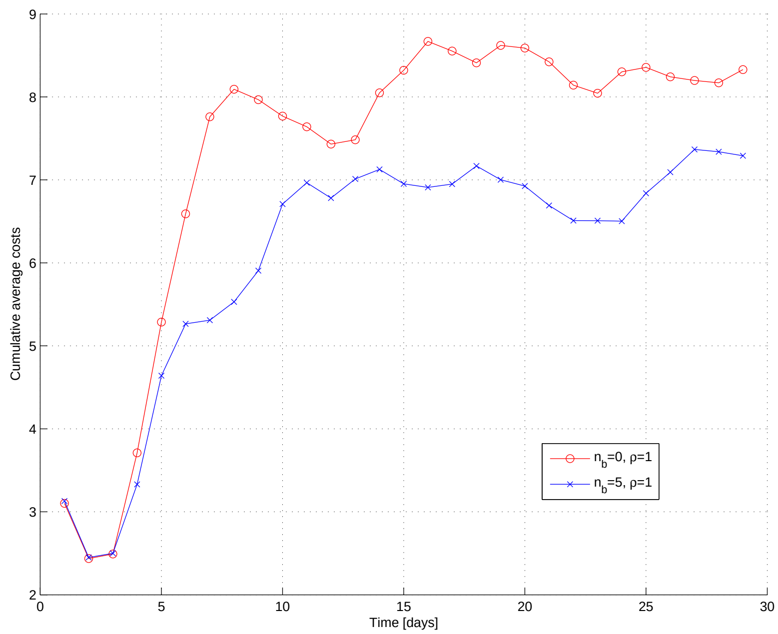

Fig. 1. Comparison of fast and slow refining.

8 Conclusions and Further Work

In this paper, we have addressed the trajectory optimization problem for cases when the prior knowledge about the problem is limited. We offered an approach based on simplification in both actions (using a set of simplification s) and states (using first PCA factors). The theoretical

approach has been illustrated in a case study related to optimal cooling in a building. It has been demonstrated that slow refining leads to better results than addressing the high-dimensional problem directly.
The promising results challenge further research. From the theoretical point of view, more precise statistical tests for next refining shall be established. Next, the applicability of alternative black-box models and optimization algorithms shall be evaluated. Then, more extensive tests shall provide more evidence about the benefits of the proposed methodology, especially on comparison to model predictive control. Finally, the approach might be applied also in another domains, such as inventory management.

References

1. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley & sons (2007)
2. Rojas, R.: Neural Networks - A Systematic Introduction. Springer-Verlag, Berlin (1996)
3. Xu, L., Jordan, M.: On convergence properties of the em algorithm for gaussian mixtures. Neural Computation **8**(1) (1996) 129–151
4. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. MIT Press (2006)
5. Cleveland, W.: Robust locally weighted regression and smoothing scatterplots. American Statistical Association **74**(368) (1979) 829–836
6. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization. Journal for Global Optimization **11**(341-359) (1997)
7. Kennedy, J., Eberhart, R.C.: Particle swarm optimization, Piscataway, NJ, IEEE Int. Conf. on Neural Networks (1995) 1942–1948
8. Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: Evolutionary Computation, 1996., Proceedings of IEEE International Conference on, IEEE (1996) 312–317
9. Tvrdík, J.: Adaptation in differential evolution: A numerical comparison. Applied Soft Computing **9**(3) (2009) 1149 – 1155
10. Macek, K., Boštík, J., Kukal, J.: Reinforcement learning in global optimization heuristics. In: Mendel, 16ts International Conference on Soft Computing. (2010) 22–28
11. Zhou, Q., Wang, S., Ma, Z.: A model-based fault detection and diagnosis strategy for hvac systems. International Journal of Energy Research **33** (2010) 903–918
12. Henze, G., Kalz, D., Liu, S., Felsman, C.: Experimental analysis of model based predictive optimal control for active and passive building thermal storage inventory. HVAC&R Research **11**(1) (2004) 183–213
13. Střelec, M., Macek, K., Abate, A.: Modeling and simulation of a microgrid as a stochastic hybrid system. In: Proceedings of the IEEE PES Innovative Smart Grid Technologies (ISGT 12). (2012)
14. Wasserman, L.: All of Nonparametric Statistics. Springer (2006)
15. Jolliffe, I.: Principal Component Analysis. Springer Verlag (1986)