Alternating between Surrogate Model Construction and Search for Configurations of an Autonomous Delivery System

Chin-Hsuan Sun
National Taiwan University
Taipei, Taiwan
r10945004@ntu.edu.tw

Thomas Laurent
National Institute of Informatics
JSPS International Research Fellow
Tokyo, Japan
thomas-laurent@nii.ac.jp

Paolo Arcaini
National Institute of Informatics
Tokyo, Japan
arcaini@nii.ac.jp

Fuyuki Ishikawa
National Institute of Informatics
Tokyo, Japan
f-ishikawa@nii.ac.jp

***Abstract*—Autonomous robots are emerging as a solution to various challenges of last mile goods delivery, like reducing traffic congestion, pollution, and costs. The configuration of an autonomous delivery robots system requires balancing aspects like delivery rate, cost of robots' operation, and required monitoring efforts. Our industry partner Panasonic is employing a search-based approach to find the configurations of the system that optimise these three aspects for a given set of customers' orders. The approach uses a simulator to assess the different configurations in the fitness functions' computation. Due to the high cost of the simulation, the whole search-based approach is computationally expensive. A classic approach to speed up such approaches is to use surrogate models trained on example simulation data that allow to approximate the results of a simulated configuration with negligible computational cost. A risk when using such approaches is to underestimate the cost of building the surrogate model itself, that can exceed the computational gain obtained during the search, thus making the adoption of surrogate models detrimental. In this work, we propose an approach in which the surrogate model is not trained before the search; instead, the approach alternates between training the model on subsets of data of increasing size, and searching using these cheaper models until the search stagnates. Experiments over 144,000 settings of the search show that the proposed approach can significantly reduce the cost of searching for configurations, while having an acceptable impact on the quality of the configurations it finds.**

***Index Terms*—surrogate models, model refinement, search-based configuration, autonomous robots, goods delivery**

I. Introduction

Goods delivery services have seen a recent boom, especially in the context of the COVID-19 pandemic [1]. For these services, *last mile delivery* – between the store (e.g., supermarket) and the client – presents a series of challenges and leads to several problems such as traffic congestion [2], pollution [3], and high operational costs. On top of these, some contextual difficulties, such as labour shortage in Japan [4], can further hinder the operation of these services.

Automation, in the form of *autonomous delivery robots*, is emerging as a solution to some of these issues. These robots lower the cost of operation in last mile delivery, while having lower impact on traffic and the environment than delivery using classic vehicles. METI, the Ministry of Economy, Trade and Industry of Japan is thus supporting public-private joint initiatives to assess the potential impacts of these solutions [5].

In this line of research, Panasonic is trialling an *autonomous delivery system* relying on autonomous robots in the Fujisawa Sustainable Smart Town in Japan [6], [7]. A central operation centre receives online orders from residents of the town to a local store, and dispatches robots to collect and deliver these orders for next-day delivery. Through this trial, Panasonic wants to understand the resources needed by the system, mainly the number of robots and their operating hours. On the one hand, producing and deploying robots represents a cost to the company; on the other hand, providing the system with insufficient resources will lead to orders not being delivered and, therefore, low customer satisfaction.

In order to find deployment configurations of the system that are both efficient and effective (providing quality service at a low cost), Panasonic is relying on a search-based approach [8], [9] (called SIM in this paper) that uses a simulator to assess different configurations of the system, in terms of quality of the service, cost of operating the robots, and cost of monitoring the robots. The delivery system is a complex problem, that involves multiple orders and robots, complex road layouts, and interactions with other road users. Simulating such a system

P. Arcaini and F. Ishikawa are supported by Engineerable AI Techniques for Practical Applications of High-Quality Machine Learning-based Systems Project (Grant Number JPMJMI20B8), JST-Mirai. We thank our industry partner Panasonic for providing the software and the simulator used in this work, and for the discussing the principles of configuring complex real-world autonomous delivery systems.

is non-trivial and takes a significant amount of time (in the order of minutes). This computational cost of the simulation is accumulated in SIM, which requires the simulation of many candidate configurations. Since Panasonic needs to run SIM to find the best configuration for different patterns of customers' orders, the time needed to run the approach becomes an issue.

A classic approach to speed up computationally expensive search-based approaches is to use *surrogate models* [10], [11], [12] that are able to approximate the values of the fitness function(s), but are much faster to execute. However, an issue with surrogate models is that, in order to be trained, they require sample simulations of the modelled system, and these have a cost. Therefore, the cost of building the surrogate model itself is non-negligible and, if too many samples are used for the training, this cost can exceed the benefit that is obtained during the search, making its adoption not necessary and, actually, detrimental.

Contribution. In this paper, we aim at adopting surrogate models in SIM but, at the same time, minimising the cost of training them. To do so, we start from observing that, during the search, it is not always necessary to have very precise models that are costly to train. Indeed, in early generations of the search in which candidate solutions are spread over the search space, it is sufficient to have a model that is able to roughly discriminate between good and bad solutions; instead, in later generations in which candidate solutions are closer to each other, a more precise model may be needed.

Starting from this observation, we propose an approach, called SURR, that alternates between the training of the surrogate model on increasing subsets of training data and the search using it. As the search progresses, the model is trained on more and more data, making it more accurate. The search stops when it converges, i.e., candidate solutions stop improving. Since this can happen before all the training data is used, there is a potentially significant saving in terms of computational time.

We implemented SURR and applied it to Panasonic's system, using the same problem setting used to assess SIM [8]. We compared the two approaches in 144,000 different settings. Results show that SURR can significantly reduce the cost of the search, i.e., the number of simulations run. However, as expected of a surrogate model based approach, it also leads to a decrease in the performance of the search.

Paper structure. Sect. II first introduces the context necessary to the understanding of this work. Then, Sect. III introduces our proposed approach. Sect. IV details the research questions explored in this work and the experiments that answer them. Sect. V describes and discusses the results of these experiments. Sect. VI examines threats to the validity of the approach and steps taken to alleviate them. Finally, Sect. VII reviews related work, and Sect. VIII concludes the paper by outlining future work.

Fig. 1: An autonomous robot delivering goods (from [6])

II. PRELIMINARIES

This section introduces background concepts and notations relevant to this work. First, in Sect. II-A, it presents Panasonic's system and the problem that needs to be solved, and then, in Sect. II-B, it introduces the search-based solution currently in use that relies on simulation.

A. Problem description

In this paper, we consider the system provided by our industrial partner Panasonic. It consists of an *autonomous delivery system* in which *customers* of a residential area order goods from a local store, and these are delivered by *autonomous robots* (such as the one shown in Fig. 1). Specifically, the system works as follows:

- customers submit online *orders*, asking for a particular product to be delivered to their home; all orders are delivered the day after they are made;
- the orders are collected by a central operation centre that dispatches autonomous robots to deliver them;[1]
- when assigned to an order, a robot picks the ordered product from the store and delivers it to the customer's house;
- a *human operator* remotely monitors the driving of the robot and, when the robot approaches a dangerous situation (e.g., too close to a pedestrian), takes some safety actions like stopping the robot and/or performing some evading manoeuvre.

The autonomous delivery service is available during a *service interval* $SI = [si_s, si_e]$. For example, Panasonic is considering to operate the service between [9:00, 14:00]. D^{SI} is the *service interval duration* in number of hours, i.e., $D^{SI} = si_e - si_s$. Each robot can operate for a fraction of the service interval.

Panasonic is currently in the evaluation phase of the service in which it must assess the performance of different *configurations* of the system. A *configuration* $conf$ consists of:

[1]The dispatching is done using an optimisation-based scheduler. Note that our work does not target the scheduler, but the configuration of the robots that are scheduled.

- $\#robs$: the number of robots available;
- $\{[at_s^i,\ at_e^i]\}_{0 \leq i < \#robs}$: the time interval in which each robot is deployed, called the robot's *available time*;
- $RobSpeed$: the speed at which robots can travel.

Panasonic is assessing how different configurations affect three metrics that are of interest to the stakeholders of the system, i.e., the store and the customers using the service, the municipality of Fujisawa, and Panasonic itself. The three metrics are:

- DO: percentage of delivered orders; maximising DO is of interest to all the stakeholders, as it directly measures the quality of the delivery service;
- UR: utilisation of the robots during the service interval; Panasonic is interested in maximising the utilisation of each robot, as unused robots constitute wasted resources;
- OI: number of human interventions. This is particular relevant for the municipality where the service is in operation, as a higher OI means that the robots were involved in different critical cases, e.g., critical interaction with humans. Indeed, if such cases are too frequent, the social acceptance of the service could diminish.

B. SIM – *Search-based configuration using simulator*

To assess different configurations of the system, Panasonic developed a simulator that allows to specify a configuration $conf$ as defined in Sect. II-A, and a set of customers' orders $Ords$. Given these inputs, the simulator simulates the whole delivery service and reports as output which orders have been delivered and by which robot.

The goal of Panasonic engineers is, for a given set of customers' orders $Ords$, to find the configuration $conf$ that optimises the three goals reported in Sect. II-A. Since manually finding the best configuration is challenging, in [8], we proposed a search-based approach to do it. Given that the approach is based on the use of the simulator in each fitness evaluation, we call it SIM in this paper, to distinguish it from the approach that we will propose in Sect. III that is based on the use of surrogate models. In the following, we describe how SIM works.

1) Individual definition: The approach adopts a population-based search (specifically, NSGA-II [13]), where *populations* of *individuals* are evolved during *generations*. A *search individual* determines the available time of each possible robot and the maximum speed of all the robots; namely, the *search variables* $\overline{x}$ are:

$$\overline{x} = [x^1_{at_s}, x^1_{wh}, \ldots, x^{maxRobs}_{at_s}, x^{maxRobs}_{wh}, x_{spd}]$$

where:

- $maxRobs$ is the maximum number of robots that can be deployed for the service; this is determined by Panasonic in advance on the basis of monetary and production constraints;
- $x^i_{at_s}$ is the starting hour of the available time of robot rob_i, with $x^i_{at_s} \in \{si_s, \ldots, si_e - 1\}$;
- x^i_{wh} is the number of working hours of robot rob_i, with $x^i_{wh} \in \{0, \ldots, si_e - si_s\}$. Setting $x^i_{wh} = 0$ means not selecting robot rob_i; this mechanism is used to decide the number of robots. The ending hour of the available time is $x^i_{at_e} = \min(x^i_{at_s} + x^i_{wh}, si_e)$;
- x_{spd} is the maximum speed of all the robots.

An *individual* $\overline{v}$ is a concrete assignment to variables $\overline{x}$:

$$\overline{v} = [v^1_{at_s}, v^1_{wh}, \ldots, v^{maxRobs}_{at_s}, v^{maxRobs}_{wh}, v_{spd}] \quad (1)$$

For a given individual $\overline{v}$, the number of robots that have been selected for the service is determined as follows:

$$\#robs(\overline{v}) = |\{i \in \{1, \ldots, maxRobs\} \mid v^i_{wh} \neq 0\}|$$

Given an individual $\overline{v}$ and a set of customers' orders $Ords$, we can simulate the service and obtain the following results:

$$\texttt{sim}(\overline{v}, Ords) = \left\langle DO^{\overline{v}}, UR^{\overline{v}}, OI^{\overline{v}} \right\rangle$$

where:

- $DO^{\overline{v}}$ is the percentage of customers' orders in $Ords$ that have been delivered;
- $UR^{\overline{v}} = \{ur^{\overline{v}}_1, \ldots, ur^{\overline{v}}_{\#robs(\overline{v})}\}$ is the *utilisation rate* of the robots; namely, $ur^{\overline{v}}_i$ is the percentage of time that the selected robot rob_i was occupied during its available time;
- $OI^{\overline{v}}$: number of interventions of the human operator.

2) Fitness functions: The fitness functions reflect the three metrics of interest for the stakeholders (see Sect. II-A), that are the objectives in the search.

The first objective is maximising the number of delivered orders:

$$f_{del}(\overline{v}) = DO^{\overline{v}} \quad (2)$$

The second objective is maximising the average utilisation of the robots:

$$f_{util}(\overline{v}) = \sum_{ur \in UR^{\overline{v}}} \frac{ur}{\#robs(\overline{v})} \quad (3)$$

The third objective is minimising the number of interventions of the human operators:

$$f_{humOp}(\overline{v}) = OI^{\overline{v}} \quad (4)$$

III. PROPOSED APPROACH

SIM, the approach presented in Sect. II-B, uses a simulator to evaluate the fitness values of each individual. Since simulating the system is expensive (a simulation can take minutes), the scalability of the approach is limited; for example, a search with a population size of 12 individuals and 20 generations can take up to 12 hours. Since Panasonic needs to find the best configuration for different sets of customers' orders, the scalability of the approach becomes an issue.

In order to tackle this problem, a common solution adopted in search-based approaches is to use *surrogate models* [10], [11], [12] that either approximate the output of the system from which the fitness function(s) is(are) computed, or directly the fitness value(s). Therefore, we decided to investigate the integration of surrogate models in SIM, with the goal of reducing the computation time of the approach.

When we started using these models, we made three observations:

- **Observation 1**: building the surrogate model has a non-negligible cost. Indeed, training the surrogate model requires running simulations of the system. If the number of simulations used in training is the same (or higher) than that used in SIM, using a surrogate model is not necessary and, actually, counterproductive;
- **Observation 2**: a surrogate model can be trained with different amounts of data, which determines its accuracy. Using more data allows for a better model but, of course, takes more time;
- **Observation 3**: the search in early generations does not need a very precise model, but one able to roughly discriminate among good and bad individuals. Indeed, in early generations, the individuals are evenly distributed over the search space and even an imprecise model is able to sort most of them correctly. Instead, as the search progresses, it finds better individuals that are close to each other in the search space; in this case, a better (i.e., more precise) model is needed to correctly discriminate among individuals.

Based on the previous observations, we propose a search-based approach, called SURR, based on surrogate models that, in some cases, can save training time.

The approach consists in alternating between surrogate models construction and search-based configuration (as described in Sect. II-B).

Sect. III-A describes how we build surrogate models, and Sect. III-B introduces the proposed approach that alternates between model construction and search.

A. Surrogate model construction

In this work, we use surrogate models $\mathcal{M}_{surr}$ that take as input a configuration of the system as encoded in a search individual $\overline{v}$ (see Eq. 1) and give as output the prediction of the three fitness functions of the search (i.e., Eqs. 2, 3, and 4), that, in SIM, are computed using the simulator.

In order to build the models, we use a dataset $\mathcal{D}$ obtained by uniformly sampling the model input space. We set the size of $\mathcal{D}$ to the number of simulations that would be required to run the search with SIM (using the maximum number of allowed generations $MaxG_{sim}$); the motivation for this choice is that using more simulations would make the model-based approach more expensive than SIM.

We will build the models starting from subsets $\mathcal{D}_p$ of $\mathcal{D}$, with $p \in (0, 100]$. Specifically, we build n subsets $Subs_{\mathcal{D}} = [\mathcal{D}_{p_1}, \ldots, \mathcal{D}_{p_n}]$ such that $p_i < p_{i+1}$ and $\mathcal{D}_{p_i} \subset \mathcal{D}_{p_{i+1}}$ (with $i \in \{1, \ldots, n-1\}$).

Fig. 2 shows the model architecture, which contains four hidden layers. The activation function applied to the output of each hidden layer is the Rectified Linear Unit (ReLU), increasing essential nonlinear features into the neural network. For model training, we employ the nested k-fold cross-validation approach [14] to determine hyperparameters of the model, such as learning rate and batch size.

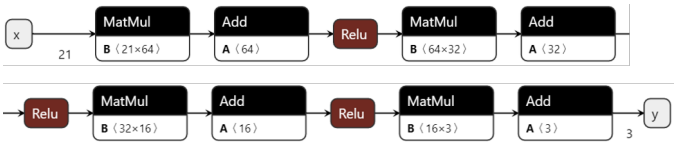


Fig. 2: Surrogate model architecture $\mathcal{M}$

Algorithm 1 SURR$_{MF}$ – Alternated model construction and search-based configuration

Require: $\mathcal{M}$: model architecture to train the $\mathcal{M}^{p\%}_{surr}$ models
Require: $Subs_{\mathcal{D}}$: list $[\mathcal{D}_{p_1}, \ldots, \mathcal{D}_{p_n}]$ of training data subsets
Require: $MaxG_{sim}$: search budget (in generations) for the simulator-based search SIM
Require: MF: multiplicative factor
Require: $PopSize$: the size of the population
Require: Th_{del}: threshold for f_{del}
Require: Th_{util}: threshold for f_{util}
Require: Th_{humOp}: threshold for f_{humOp}
1: $i \leftarrow 1$
2: Train $\mathcal{M}^{p_i}_{surr}$ with data $\mathcal{D}_{p_i}$
3: $G \leftarrow 0$
4: $Gen_{surr} \leftarrow \texttt{round}\left(\frac{|\mathcal{D}_p|}{PopSize}\right) - 1$
5: $Gen_{switch} \leftarrow MF \cdot Gen_{surr}$
6: $H \leftarrow \emptyset$
7: $MaxG \leftarrow MF \cdot MaxG_{sim}$
8: **while** $(G \leq MaxG) \wedge \neg stop\left(\begin{array}{l} H, Th_{del}, \\ Th_{util}, Th_{humOp} \end{array}\right)$ **do**
9: $H \leftarrow$ Perform one search generation using $\mathcal{M}^{p}_{surr}$
10: $G++$
11: **if** $G = Gen_{switch}$ **then**
12: Train $\mathcal{M}^{p_{i+1}}_{surr}$ with data $\mathcal{D}_{p_{i+1}}$
13: $i++$
14: $Gen_{surr} \leftarrow \texttt{round}\left(\frac{|\mathcal{D}_p|}{PopSize}\right) - 1$
15: $Gen_{switch} \leftarrow MF \cdot Gen_{surr}$
16: **end if**
17: **end while**

B. SURR – Alternating model construction and search-based configuration

In this section, we present our proposed approach (called SURR) that alternates between model construction and search-based configuration. Alg. 1 details the approach.

The approach takes in input: a model architecture $\mathcal{M}$; the list of training data subsets $Subs_{\mathcal{D}}$; the maximum number of generations $MaxG_{sim}$ that can be executed by the simulation-based approach SIM; a multiplicative factor MF that is used to determine the budget of the approach; the population size $PopSize$; and thresholds Th_{del}, Th_{util}, and Th_{humOp} that are used to decide when to stop the search.

The approach first trains model $\mathcal{M}^{p_1}_{surr}$ using data $\mathcal{D}_{p_1}$ (line 2). The training of the model requires to run $|\mathcal{D}_{p_1}|$ simulations to obtain the simulation results. This is the most expensive part of the construction the model.

In order to ensure that the use of the surrogate model

Algorithm 2 Stopping criterion *stop*

Require: Th_{del}: threshold for f_{del}
Require: Th_{util}: threshold for f_{util}
Require: Th_{humOp}: threshold for f_{humOp}
Require: W: window size
Require: H: history archive of the search
1: **if** $|H| \leq W$ **then**
2: **return** *false*
3: **end if**
4: $(max^G_{del}, max^G_{util}, min^G_{humOp}) \leftarrow$ best value of f_{del}, f_{util}, and f_{humOp} in H
5: $(max^W_{del}, max^W_{util}, min^W_{humOp}) \leftarrow$ best value of f_{del}, f_{util}, and f_{humOp} in H ignoring W last generations
6: **return** $\frac{max^G_{del} - max^W_{del}}{max^W_{del}} < Th_{del} \wedge$ $\frac{max^G_{util} - max^W_{util}}{max^W_{util}} < Th_{util} \wedge$ $\frac{min^W_{humOp} - min^G_{humOp}}{min^W_{humOp}} < Th_{humOp}$

balances out the cost of its training, the model must be used during the search more than $|\mathcal{D}_{p_1}|$ times. Therefore, we compute how many generations of SIM are executed by using $|\mathcal{D}_{p_1}|$ simulations (variable Gen_{surr} line 4). Then, at line 5, we decide until which generation Gen_{switch} of the search we use $\mathcal{M}^{p_1}_{surr}$; this is done by multiplying Gen_{surr} by an integer multiplicative factor MF (with $MF \geq 2$). The idea is that, the higher MF, the longer we use $\mathcal{M}^{p_1}_{surr}$, and so the more we capitalise on the cost of training the model. We will identify the approach SURR parameterised with an extended factor $MF = i$ as SURR_i.

Then, the approach performs the following actions:

- it runs one generation of the search using the current surrogate model $\mathcal{M}^{p_i}_{surr}$ and collects the history in H (line 9);
- then, it checks if the current generation G is equal to Gen_{switch}, i.e., the generation in which a more refined model must be trained. If this is the case:
  - the new surrogate model $\mathcal{M}^{p_{i+1}}_{surr}$ is obtained by continuing training model $\mathcal{M}^{p_i}_{surr}$ with data $\mathcal{D}_{p_{i+1}}$ (line 12); note that this step requires to first simulate the samples $\mathcal{D}_{p_{i+1}} \setminus \mathcal{D}_{p_i}$, i.e., the samples that have not been simulated for the previous model;
  - Gen_{surr} and Gen_{switch} are updated (lines 14-15) as explained before.

Before each iteration, the approach checks whether the search must stop, either because the maximum number of generations $MaxG$ is reached, or the stopping criterion *stop* is satisfied (line 8). The definition of the criterion is shown in Alg. 2. The general idea of the criterion is that the search should be stopped if there is no significant improvement for some generations. The algorithm takes as input thresholds Th_{del}, Th_{util}, and Th_{humOp} that identify the minimum improvement of each objective function that is considered significant; and a *window* W that specifies a number of generations over which to check the improvement. The criterion is checked only if the search executed at least W generations, otherwise the algorithm returns *false* (lines 1-2). If the number of current generations is higher than the window, at line 4, the approach retrieves the best value of each fitness function across all the individuals produced during the search; at line 5, it retrieves the best value of each fitness function W generations before the current one. If the relative improvement of each fitness function f is lower than a threshold Th_f[2], the stopping criterion is *true*, otherwise is *false* (line 6).

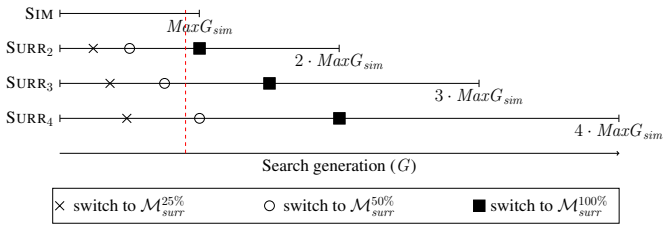


Fig. 3: Comparison of number of generations and budget (in terms of number of simulations) between SIM, SURR_2, SURR_3, and SURR_4

Remark 1. Note that, in the model construction, the most expensive part is the simulation of the sampled data, and the other costs of the training are negligible. Similarly, the cost of running the search in SURR is also negligible. Therefore, when assessing the cost of an approach, we consider the number of required simulations. Specifically, for SIM, the cost is given by the number of evaluated individuals when the search is stopped (as the simulator is called once for each individual). For SURR, instead, the cost is given by the number of simulations required to train the model used by search at the moment of stopping.

Example 1. Fig. 3 visualises the effect of using multiplicative factors $MF \in \{2, 3, 4\}$. We notice that the higher MF, the higher the maximum number of generations is. Moreover, for the same number of generations, the cost (in terms of used simulations) is different across the different approaches. For example, assume to have $PopSize = 12$ and $MaxG_{sim} = 20$; at search generation $G = 18$ (red dashed vertical line), SIM has used 228 simulations (i.e., $(18+1)\cdot 12$), SURR_2 and SURR_3 have both used $|\mathcal{D}_{p_{50\%}}| = 126$ simulations, and SURR_4 has used $|\mathcal{D}_{p_{25\%}}| = 63$ simulations.

IV. EXPERIMENT DESIGN

This section introduces the research questions we explore in this work and the experiments we conducted to answer them.

The code and the experimental results are available online [15]; for IP protection, we cannot share the simulator of the system nor the trained models.

A. Research questions

This work explores two research questions in order to assess the proposed approach SURR:

[2]Note that the fitness improvement is defined differently for the two maximisation functions than for the minimisation one.

RQ1: What is the impact of the proposed approach SURR on the cost of the search?

This research question focuses on the cost of using SURR, i.e., the number of simulations required, compared to performing the search using the simulator as in SIM. It is divided into two sub-RQs:

RQ1.1: Overall, does SURR require fewer simulations than SIM?

RQ1.2: What is the impact of the multiplicative factor MF on the cost of SURR?

RQ2: What is the effectiveness of the proposed approach SURR?

This research question analyses the quality of the solutions found by SURR. It is divided into two sub-RQs:

RQ2.1: Overall, how do the solutions found by SURR and by the simulation-based approach SIM compare with each other?

RQ2.2: What is the impact of the multiplicative factor MF on the effectiveness of SURR?

B. Setting of the delivery system

The search-based approaches used in this paper (SIM and SURR) aim at solving the configuration problem described in Sect. II-A. The problem consists in finding the best configurations of the robots for a given service interval duration D^{SI}, set of customers' orders $Ords$, and maximum number of robots $maxRobs$. For the experiments, we use the same setting used in [8] in which SIM was proposed: $D^{SI} = 5$ hours; $Ords$ as a set of representative 50 customers' orders; and $maxRobs = 10$.

C. Compared approaches

The first approach considered in the experiments is SIM (see Sect. II-B) that uses the simulator in the fitness functions' evaluation. In this approach, the search algorithm used is NSGA-II [13] with population size $PopSize = 12$, offspring set to 12, and budget in terms of maximum number of generations $MaxG_{sim} = 20$. With this setting, if SIM is executed until $MaxG_{sim}$, it must run $(20 + 1) \cdot 12 = 252$ simulations. Regarding the search operators, we used the default settings of NSGA-II in jMetalPy [16], as it has been shown that default settings can provide reasonable results [17]: parents selection with Binary tournament, SBX crossover operator, crossover rate of 0.9, polynomial mutation operator, and mutation rate equal to the reciprocal of the number of variables.

The second considered approach is SURR, the one we propose in this work (see Sect. III). The parameters of the search are kept the same as in SIM, except for the maximum number of generations. Indeed, the maximum number of generations $MaxG$ is computed as $MaxG = MF \cdot MaxG_{sim}$ (see line 7 in Alg. 1); we consider $MF \in \{2, 3, 4\}$, that correspond to three versions of the approach, namely SURR_2, SURR_3, and SURR_4.

As described in Sect. III, SURR builds different models in an incremental fashion. To do this, it requires different subsets $Subs_{\mathcal{D}} = [\mathcal{D}_{p_1}, \ldots, \mathcal{D}_{p_n}]$ of training data. For the experiments, we use four subsets $\mathcal{D}_{12.5\%}$, $\mathcal{D}_{25\%}$, $\mathcal{D}_{50\%}$, $\mathcal{D}_{100\%}$ that respectively use 12.5%, 25%, 50%, and 100% of the available training data $\mathcal{D}$. Table I reports the concrete number of simulations needed for training each of the four models $\mathcal{M}_{surr}^{12.5\%}$, $\mathcal{M}_{surr}^{25\%}$, $\mathcal{M}_{surr}^{50\%}$, and $\mathcal{M}_{surr}^{100\%}$. Moreover, it also reports how many generations of SIM can be run using those numbers of simulations (i.e., Gen_{surr} computed at lines 4 and 14 in Alg. 1).

TABLE I: Surrogate models $\mathcal{M}_{surr}^{p}$ – Cost of training the model and corresponding value of Gen_{surr}

Model	$\lvert\mathcal{D}_p\rvert$ (# simulations used to train)	Gen_{surr} (equiv. # of search generations using SIM)
$\mathcal{M}_{surr}^{12.5\%}$	32	2
$\mathcal{M}_{surr}^{25\%}$	63	4
$\mathcal{M}_{surr}^{50\%}$	126	10
$\mathcal{M}_{surr}^{100\%}$	252	20

TABLE II: Gen_{switch} – Generation until which a surrogate model is used

	$\mathcal{M}_{surr}^{12.5\%}$	$\mathcal{M}_{surr}^{25\%}$	$\mathcal{M}_{surr}^{50\%}$	$\mathcal{M}_{surr}^{100\%}$
SURR_1	2	4	10	20
SURR_2	4	8	20	40
SURR_3	6	12	30	60
SURR_4	8	16	40	80

The approach uses a surrogate model $\mathcal{M}_{surr}^{p}$ until generation Gen_{switch}, that is computed using Gen_{surr} and MF (see lines 5 and 15 in Alg. 1). Table II reports, for each version of the approach, the value of Gen_{switch} for each model $\mathcal{M}_{surr}^{p}$.

Stopping criterion: All approaches (SIM, and the three versions of SURR) use the stopping criterion $stop$ described in Alg. 2. In order to limit the effect of the chosen thresholds and window values on the performance of the different approaches, we instantiate all approaches with 144,000 settings of the criterion. These settings correspond to all the combinations of $stop$'s parameters set at follows: Th_{del} between 0.1% and 10%, sampling 30 values uniformly; Th_{util} between 0.1% and 10%, sampling 30 values uniformly; and Th_{humOp} between 10% and 50%, sampling 20 values uniformly; and W taking all values between 3 and 10. These values were obtained empirically by ensuring they produce criteria that neither stop the search prematurely or never (for any approach).

D. Experimental setup

An *experiment run* consists in the execution of one approach (SIM, SURR_2, SURR_3, and SURR_4) using a given stopping criterion. In order to account for the randomness involved in the approaches, each run of the experiment was performed 30 times [18]. All runs of SURR use the same datasets, but perform separate training of the surrogate models.

E. Evaluation metrics

Metrics for RQ1: In order to answer RQ1 related to the computational cost of an approach, we consider the number of

simulations $usedSims$ that have been executed, as the cost of training and inference with the surrogate model, and the cost of the search operations (i.e., mutation, crossover, etc.) are negligible compared to the cost of simulation (see discussion in Remark 1). This number is computed by considering the number of simulations used at generation G_{stop} when the search stopped (either by reaching $MaxG_{sim}$ or by triggering $stop$). For SIM, $usedSims$ is given by the number of evaluated individuals (as each individual requires one simulation). For SURR, we consider which surrogate model $\mathcal{M}^p_{surr}$ was in use at generation G_{stop}; $usedSims$ corresponds to the number of simulations used to train $\mathcal{M}^p_{surr}$, as reported in Table I.

To compare two approaches with a given stopping criterion we compare the distribution of the cost $usedSims$ over the 30 runs, using proper statistical tests as recommended by Arcuri and Briand [18]. We first apply the Mann-Whitney U test to determine if the two distributions are statistically significantly different. If the p-value returned by the test is lower than the confidence value α (with $\alpha = 0.05$ in these experiments), we reject the null hypothesis and consider that there is a significant difference between the two distributions. If the distributions are significantly different, we apply Vargha and Delaney's $\hat{A}_{12}$ effect size [19] to determine the strength of the significance, using the categories proposed by Kitchenham et al. [20]: *negligible* when $\hat{A}_{12} \in (0.494, 0.5)$, *small* when $\hat{A}_{12} \in (0.362, 0.494]$, *medium* when $\hat{A}_{12} \in (0.286, 0.362]$, and *large* when $\hat{A}_{12} \leq 0.286$. The same categories are defined symmetrically when $\hat{A}_{12} > 0.5$.

To better gauge the difference between two approaches, we also compute the average savings in number of generations provided by the method over all the stopping criteria for which it is statistically significantly cheaper with at least medium strength in terms of $\hat{A}_{12}$.

Metrics for RQ2: In order to answer RQ2, we consider the *quality* of the sets of configurations produced by the different approaches; all the approaches considered in this work are multi-objective and thus return a Pareto front of solutions rather than a single configuration. In order to reflect the quality of Pareto fronts, different *quality indicators* are available [21], that return a single numerical value that represents the Pareto front. For our experiments, we select the Inverted Generational Distance (IGD), as suggested by Ali et al. [22] for evaluating solutions produced by NSGA-II. In order to compute the IGD at generation G, we build a Pareto front from all individuals produced up G. Smaller values of IGD reflect a better quality of the Pareto front in the objective space. Note that all solutions of SURR are simulated before computing the IGD, making sure that they are compared fairly in the objective space, i.e., without the errors introduced by the surrogate model-based fitness evaluation. We apply the same statistical approach to compare two approaches with a given stopping criterion as in RQ1 and also compute the average gain in IGD provided by a method when it produces better configurations.

V. EXPERIMENTAL RESULTS

This section reports and discusses the results of the experiments described in Sect. IV.

A. Answer to RQ1

In this RQ, we want to assess whether the proposed approach SURR is able to save time for the search.

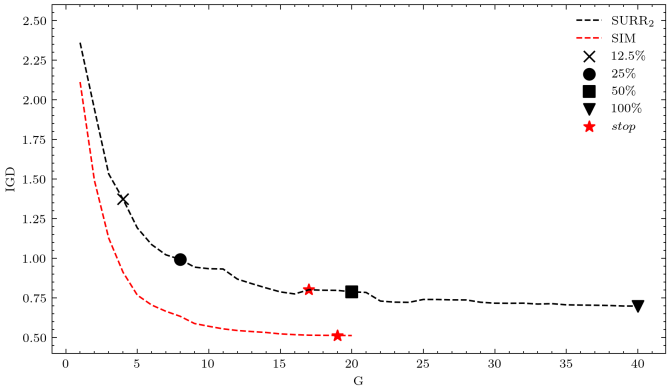
Fig. 4 visualises the overall evolution of the search, that can be used to have an initial assessment for RQ1 and also for RQ2. For each considered value of the multiplicative factor MF (in each sub-figure), it shows the quality of the configurations (in terms of IGD) produced by SURR_{MF} along the search generations G, as well as that of those produced by SIM for reference. The average generation where the search stopped across all 144,000 stopping criteria is indicated on each plot with a red star (★). The plots of SURR_{MF} also show with symbols ✕, ●, ■, and ▼, the generation until when each model is used (i.e., model $\mathcal{M}^{12.5\%}_{surr}$ is used until ✕, model $\mathcal{M}^{25\%}_{surr}$ is used until ●, etc.); this gives an indication of the average cost of each approach. Looking at these plots, we can see that SURR_2, SURR_3, and SURR_4 stop on average before using $\mathcal{M}^{100\%}_{surr}$, i.e., they have on average a lower cost than SIM. Specifically, SIM stops on average at generation 19, which corresponds to 240 simulations; SURR_2 stops at generation 17 when $\mathcal{M}^{50\%}_{surr}$ is used, which requires 126 simulations for training; in a similar way, also SURR_3 and SURR_4 stop, on average, when using $\mathcal{M}^{50\%}_{surr}$. This means that, on average, SURR allows to save half of the computation time.

Fig. 5 provides a more detailed comparison between the different approaches. It shows, for each pair of approaches, the proportion of stopping criteria for which the method on the row is statistically significantly better in terms of cost than the one on the column with at least medium strength in terms of $\hat{A}_{12}$ (see Sect. IV-E).

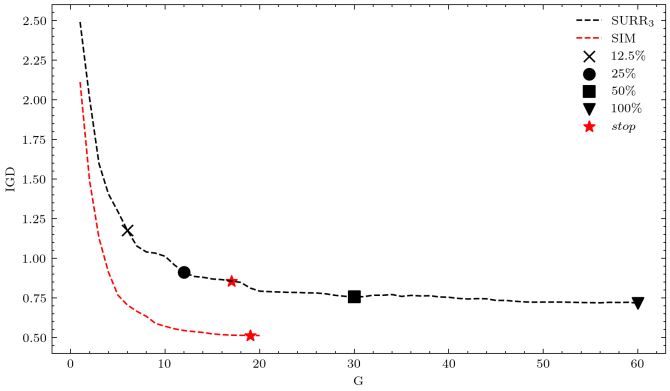
The results in the last line and in the last column of the matrix pertain to RQ1.1, and show that the simulator based approach SIM is never cheaper than SURR, and that SURR is often (for 86.9% of the stopping criteria with SURR_2 and always with SURR_3 and SURR_4) significantly cheaper than running the search using the simulator as in SIM.

The rest of the matrix is related to RQ1.2, and reflects the influence of MF on the cost of SURR_{MF}. Overall, we see that higher values of MF lead to better cost, i.e., to the search stopping after fewer training steps of the surrogate model. This can be explained by the fact that a higher MF leads to each surrogate model being used for a higher number of generations in the search, giving more opportunity for the search to stop before switching to a more expensive model. Note that the increased number of generations represents a negligible overhead in comparison to the cost of running the simulator to train the better model (see discussion in Remark 1).

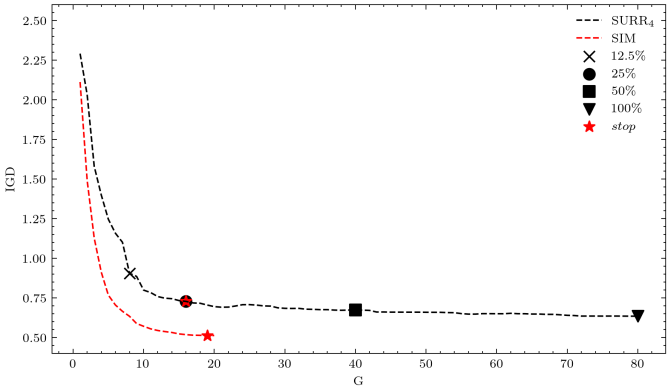
We further check the concrete saving of the approaches. Fig. 6 reports the average saving between all the approaches. Specifically, it reports, for all the stopping criteria in which the approach on the rows is statistically significantly better



(a) SURR$_2$



(b) SURR$_3$



(c) SURR$_4$

Fig. 4: RQ1 and RQ2 – Quality of Pareto fronts in terms of IGD throughout the search generations G (average of all the 144,000 stopping criteria)

than the approach on the columns (as reported in Fig. 5), the average difference in terms of used simulations. These results confirm that a higher value of MF leads to greater cost gains; for example, SURR$_4$ provides, on average, a saving of 122.12 simulations w.r.t. SIM (nearly half of the maximum number of possible simulations), and of 69.78 and 38.63 simulations w.r.t. SURR$_2$ and SURR$_3$.

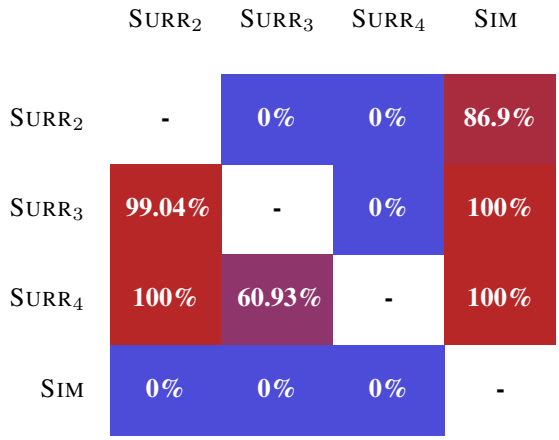


Fig. 5: RQ1 – Statistical comparison in terms of cost (number of used simulations) between all the approaches

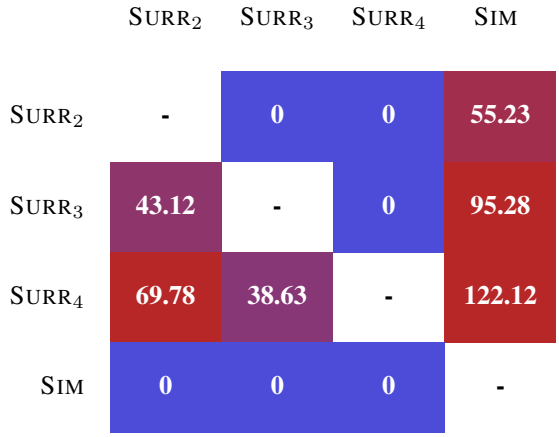


Fig. 6: RQ1 – Cost gain in terms of number of used simulations between all the approaches

Answer to RQ1.1: SURR often significantly lowers the cost of the search compared to SIM, i.e., it requires fewer simulations. SURR never requires more simulations than SIM.
Answer to RQ1.2: The multiplicative factor MF impacts the cost of SURR$_{MF}$, with higher values of MF leading to a lower cost.

B. Answer to RQ2

In this RQ, we are interested in assessing the effect of the proposed approach on the quality of the final solutions in terms of IGD.

Fig. 4 allows to make an initial assessment. By looking at the value of IGD at the average stopping point of each method (the red star ★), we can get a sense of the quality of the configurations it produces; we see that, on average, for all values of MF, SURR$_{MF}$ does not achieve an IGD as good (i.e., as low) as that achieved by SIM.

These results are confirmed in Fig. 7 that provides a more detailed comparison between the different approaches. It shows, for each pair of approaches, the proportion of stopping

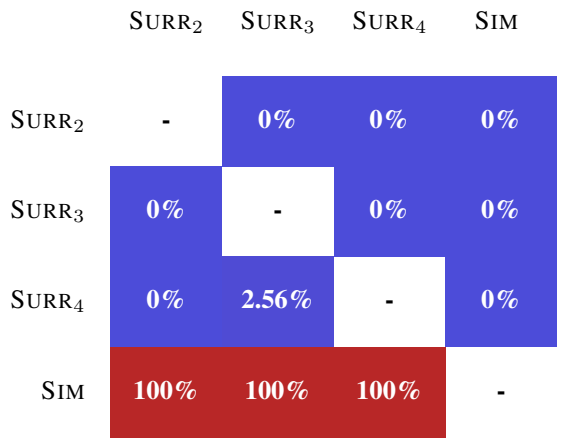


Fig. 7: RQ2 – Statistical comparison of the effectiveness in terms of IGD between all the approaches

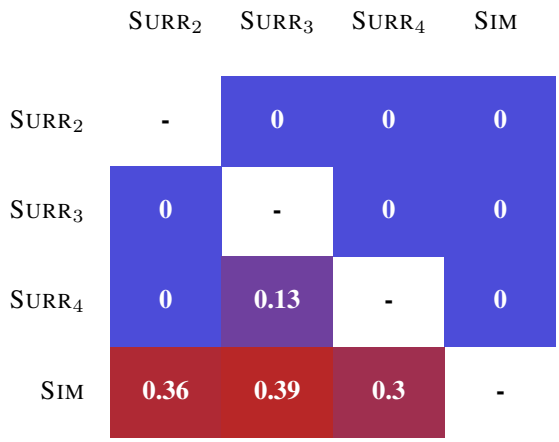


Fig. 8: RQ2 – Effectiveness gain in terms of IGD between all the approaches

criteria for which the method on the row is statistically significantly better in terms of IGD than the one on the column with at least medium strength in terms of $\hat{A}_{12}$ (see Sect. IV-E). The results in the last line and in the last column of the matrix pertain to RQ2.1, and show that, in all cases, SURR$_{MF}$ does not produce configurations of the same quality as SIM.

This observation is not surprising, and is part of the accepted trade-off when using surrogate models in search-based approaches. Indeed, when the computational cost of the approach is significantly reduced, a certain level of noise is introduced, leading to inaccuracies in the search and a final Pareto front that is not as good as the one obtained when using the original, expensive way of computing the fitness function(s). This trade-off is accepted by stakeholders of the autonomous delivery system considered in this paper, who accept to loose some effectiveness for a large gain in efficiency. Indeed, this allows to apply the search-based configuration approach at a real-world scale on a daily basis.

The rest of the matrix in Fig. 7 is related to RQ2.2, and shows that MF has nearly no influence on the effectiveness of SURR$_{MF}$. This means that we can use the approach that saves the most execution time (see results of RQ1 in Sect. V-A), without loosing in effectiveness.

We further check the concrete difference in effectiveness. Fig. 8 shows the average gains in IGD that each approach provides compared to each other. Specifically it reports, for all the stopping criteria in which the approach on the rows is statistically significantly better than the approach on the columns (as reported in Fig. 7), the difference of IGD. The results confirm that SIM produces configurations of higher quality that those produced by SURR$_{MF}$. Instead, by comparing the different versions of SURR$_{MF}$, we observe that the impact of MF is minimal.

Answer to RQ2.1: SURR introduces noise through the use of surrogate models, and thus does not produce configurations that achieve the same performance as those produced by SIM. This is balanced by the gain in cost (see RQ1) and makes the approach more scalable and applicable in a real-world context.
Answer to RQ2.2: The multiplicative factor MF has nearly no impact on the effectiveness of SURR$_{MF}$.

VI. THREATS TO VALIDITY

Different threats may affect the validity of the approach. We discuss them in the following, using the classical classification of *construct*, *conclusion*, *internal*, and *external validities* [23].

Construct validity: A typical threat to the construct validity of the approach is that the metrics that we use to assess SURR could be not suitable. Since the goal of the adoption of surrogate models is to achieve scalability, considering the number of required simulations when the approach stops is a suitable metric; indeed, the simulations are the most computational demanding aspect of the approach, while other activities have a negligible impact on computational cost. Since SURR introduces noise on the search, we also check the quality of the final solutions; since the approach is multi-objective, the typical approach to assess the quality of solutions is to use quality indicators [21]. Moreover, we used IGD that has been shown to be the most suitable indicator to assess solutions produced by NSGA-II [22].

Conclusion validity: The random behaviour of the compared approaches (both model training and evolutionary computation) must be taken into consideration in the assessment. By following the guideline of Arcuri and Briand [18], we executed each experiment of SIM and of the three versions of SURR$_{MF}$ 30 times. Moreover, still following [18], when comparing two approaches, we checked both statistical significance with Mann-Whitney U test, and effect size with $\hat{A}_{12}$ statistics. To have stronger conclusions, we considered having a significant difference only the cases showing at least medium strength for the effect size $\hat{A}_{12}$ [20].

Internal validity: An internal validity threat is that the results obtained in the experiments could be obtained by chance. This can happen due to, for example, a faulty implementation.

To mitigate this threat, we have carefully checked and tested the implementation.

External validity: A typical threat to external validity is that the alternate approach that we implemented for the search-based configuration of the Panasonic's system is not applicable for other systems. We think that, in principle, any system for which a surrogate model can be built is a suitable target of our approach, although some systems may require too many simulations for building a reliable initial model, which could make the approach not advantageous. However, as Briand et al. [24] pointed out, research driven by industry necessarily targets the specific problem of the company, and is not primarily concerned with applicability to other problems. We thus leave the investigation of generalisability to future work.

VII. Related work

This section reviews works related to the approach that we present in this paper.

Different surveys have been published on surrogate models and their adoption. Tong et al. [11] survey the use of surrogate models in evolutionary algorithms, and distinguish between *absolute fitness models* approximating the fitness function values (as in SURR), and *relative fitness models* estimating the relative rank of solutions. However, they limit their study to single-objective problems, while Panasonic's problem is a multi-objective problem.

Another survey is provided by Jin [10] that also reviews the use of surrogate models in evolutionary computation. The survey considers both single- and multi-objective optimisation problems, constrained optimisation problems, dynamic optimisation problems, and multi-modal optimisation problems.

Najati et al. [12] more specifically focus on the use of surrogate models in search-based testing [25]. They highlight the use of surrogate models to approximate complex simulated test beds and to compute fitness functions more efficiently, i.e., the way $\mathcal{M}_{surr}$ employs them. They also contrast *static approaches*, that employ a fixed surrogate model, and *dynamic approaches* that improve the model as they use it. $\mathcal{M}_{surr}$ falls into the latter category, although it is not a testing approach.

Surrogate models have been applied in different approaches for different applications. For example, Ben Abdessalem et al. [26] consider a multi-objective search problem and the application of neural networks as surrogate models to approximate fitness functions for the testing of a Pedestrian Detection Vision (PeVi) system. The approach always uses surrogate models for the fitness computation, but also re-assesses some solutions when the surrogate model's confidence is not sufficient to rank solutions reliably. Differently from SURR, the surrogate model is never refined during the search.

Similarly, Haq et al. [27] and Auer et al. [28] use surrogate models to enhance a multi-objective search-based test generation process for DNN-enabled systems and Android applications respectively. However, they only consider a fixed model during the search, while SURR refines the model as the search progresses and considers more similar individuals.

Menghi et al. [29] use surrogate models to approximate costly Cyber-Physical Systems during the falsification process [30], a type of search-based testing that tries to violate a system specification. They also employ a refinement approach for their model, however this is only triggered when a spurious failure (a failure according to the surrogate model but not with the real system) is found. In contrast, SURR systematically performs refinement along the search for configurations on specific generations.

Beglerovic et al. [31] use surrogate models to test autonomous vehicles. Similarly to SURR, the surrogate model is systematically refined during the search, but more frequently (at each iteration of the search). Differently from our approach, they consider a single-objective function, while we consider a multi-objective one.

Gambi et al. [32] propose a search-based generator for elastic systems, that uses surrogate models to speed up the search. Differently from SURR, they consider a single-objective function and the surrogate model is never refined.

Finally, Biagiola and Tonella [33] use a binary classifier to approximate the pass or fail output of Deep Reinforcement Learning agents' tests and compute a fitness function to guide the test generation process. Differently from our approach, the surrogate model is built at the beginning of the search and never refined.

Regarding the Panasonic autonomous delivery system that we consider in this paper, it has been considered in other works for finding the best configuration [8], [9], for finding the most stable configuration [34], and for testing the scheduler employed in the system [35].

VIII. Conclusion

This work introduces a search-based approach that uses surrogate models to efficiently find configurations of an autonomous delivery system operated by autonomous robots. The approach leverages the larger difference between individuals at the beginning on the search to allow for the use of simpler, cheaper surrogate models that are refined as the search progresses. Results show that this approach is much cheaper than Panasonic's current approach of running a search using only their simulator to assess individuals. However, these gains in cost are accompanied by a drop in effectiveness of the configuration generation approach, a known trade-off of surrogate model based approaches.

Results show that the multiplicative factor MF, which controls when the surrogate model is refined during the search, impacts the efficiency of the approach. However, we do not know a priori which is the most effective value of MF, or even if always using the same value of MF is the best approach. As future work, we plan to investigate a more dynamic way of triggering the refinement of the model, e.g., by considering the spread of the current population in the search space.

As another line of work, we plan to explore the combined use of the surrogate model and the simulator in the fitness computation, to maximise the cost gains while limiting the drop in effectiveness.

REFERENCES

[1] X. C. Wang, W. Kim, J. Holguín-Veras, and J. Schmid, "Adoption of delivery services in light of the COVID pandemic: Who and how long?" *Transportation Research Part A: Policy and Practice*, vol. 154, pp. 270–286, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965856421002676

[2] G. Mangano and G. Zenezini, "The value proposition of innovative last-mile delivery services from the perspective of local retailers," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2590–2595, 2019, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319315848

[3] M. Viu-Roig and E. J. Alvarez-Palau, "The impact of E-commerce-related last-mile logistics on cities: A systematic literature review," *Sustainability*, vol. 12, no. 16, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/16/6492

[4] S. Reiko, "Japan's logistics crisis," https://www3.nhk.or.jp/nhkworld/en/news/backstories/1771/, 2021.

[5] METI, "Make delivery smart with automated delivery robots," https://www.meti.go.jp/english/mobile/2022/20220802001en.html, 2022.

[6] Panasonic Holdings Corporation, "Panasonic to conduct field test of home delivery service by compact, low-speed robot in Fujisawa sustainable smart town," https://news.panasonic.com/global/press/en201214-1, December 2020, last access: October 25, 2023.

[7] M. Sakurai and J. Kokuryo, "Fujisawa sustainable smart town: Panasonic's challenge in building a sustainable society," *Communications of the Association for Information Systems*, vol. 42, no. 1, p. 19, 2018.

[8] P. Arcaini, E. Castellano, F. Ishikawa, H. Kawamoto, K. Sawai, and E. Muramoto, "Incremental search-based allocation of autonomous robots for goods delivery," in *2023 IEEE Congress on Evolutionary Computation (CEC)*, 2023, pp. 1–10.

[9] M. Byrd Victorica, P. Arcaini, F. Ishikawa, H. Kawamoto, K. Sawai, and E. Muramoto, "Stability-aware exploration of design space of autonomous robots for goods delivery," in *2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2023, pp. 177–186.

[10] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650211000198

[11] H. Tong, C. Huang, L. L. Minku, and X. Yao, "Surrogate models in evolutionary single-objective optimization: A new taxonomy and experimental study," *Information Sciences*, vol. 562, pp. 414–437, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521002395

[12] S. Nejati, L. Sorokin, D. Safin, F. Formica, M. M. Mahboob, and C. Menghi, "Reflections on surrogate-assisted search-based testing: A taxonomy and two replication studies based on industrial ADAS and Simulink models," *Information and Software Technology*, vol. 163, p. 107286, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584923001404

[13] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[14] J. Wainer and G. Cawley, "Nested cross-validation when selecting classifiers is overzealous for most practical applications," *Expert Systems with Applications*, vol. 182, p. 115222, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421006540

[15] C. Sun, T. Laurent, P. Arcaini, and F. Ishikawa, "Supplementary material for the paper "Alternating between Surrogate Model Construction and Search for Configurations of an Autonomous Delivery System"," https://github.com/ERATOMMSD/surrogate_models_delivery_robots, 2024.

[16] A. Benítez-Hidalgo, A. J. Nebro, J. García-Nieto, I. Oregi, and J. Del Ser, "jMetalPy: A Python framework for multi-objective optimization with metaheuristics," *Swarm and Evolutionary Computation*, vol. 51, p. 100598, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650219301397

[17] A. Arcuri and G. Fraser, "Parameter tuning or default values? an empirical investigation in search-based software engineering," *Empirical Software Engineering*, vol. 18, pp. 594–623, 2013.

[18] A. Arcuri and L. Briand, "A practical guide for using statistical tests to assess randomized algorithms in software engineering," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11. New York, NY, USA: ACM, 2011, pp. 1–10.

[19] A. Vargha and H. D. Delaney, "A critique and improvement of the CL common language effect size statistics of McGraw and Wong," *Journal of Educational and Behavioral Statistics*, vol. 25, no. 2, pp. 101–132, 2000.

[20] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs, and A. Pohthong, "Robust statistical methods for empirical software engineering," *Empirical Softw. Engg.*, vol. 22, no. 2, pp. 579–630, apr 2017.

[21] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Comput. Surv.*, vol. 52, no. 2, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3300148

[22] S. Ali, P. Arcaini, D. Pradhan, S. A. Safdar, and T. Yue, "Quality indicators in search-based software engineering: An empirical evaluation," *ACM Trans. Softw. Eng. Methodol.*, vol. 29, no. 2, Mar. 2020. [Online]. Available: https://doi.org/10.1145/3375636

[23] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.

[24] L. C. Briand, D. Bianculli, S. Nejati, F. Pastore, and M. Sabetzadeh, "The case for context-driven software engineering research: Generalizability is overrated," *IEEE Software*, vol. 34, no. 5, pp. 72–75, 2017.

[25] M. Harman, S. A. Mansouri, and Y. Zhang, "Search-based software engineering: Trends, techniques and applications," *ACM Comput. Surv.*, vol. 45, no. 1, Dec. 2012. [Online]. Available: https://doi.org/10.1145/2379776.2379787

[26] R. Ben Abdessalem, S. Nejati, L. C. Briand, and T. Stifter, "Testing advanced driver assistance systems using multi-objective search and neural networks," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE 2016. New York, NY, USA: ACM, 2016, pp. 63–74.

[27] F. U. Haq, D. Shin, and L. Briand, "Efficient online testing for DNN-enabled systems using surrogate-assisted and many-objective optimization," in *Proceedings of the 44th International Conference on Software Engineering*, ser. ICSE '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 811–822. [Online]. Available: https://doi.org/10.1145/3510003.3510188

[28] M. Auer, F. Adler, and G. Fraser, "Improving search-based Android test generation using surrogate models," in *Search-Based Software Engineering: 14th International Symposium, SSBSE 2022, Singapore, November 17–18, 2022, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2022, pp. 51–66. [Online]. Available: https://doi.org/10.1007/978-3-031-21251-2_4

[29] C. Menghi, S. Nejati, L. Briand, and Y. I. Parache, "Approximation-refinement testing of compute-intensive cyber-physical models: An approach based on system identification," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ser. ICSE '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 372–384. [Online]. Available: https://doi.org/10.1145/3377811.3380370

[30] C. Menghi, P. Arcaini, W. Baptista, G. Ernst, G. Fainekos, F. Formica, S. Gon, T. Khandait, A. Kundu, G. Pedrielli, J. Peltomäki, I. Porres, R. Ray, M. Waga, and Z. Zhang, "ARCH-COMP23 category report: Falsification," in *Proceedings of 10th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH23)*, ser. EPiC Series in Computing, G. Frehse and M. Althoff, Eds., vol. 96. EasyChair, 2023, pp. 151–169. [Online]. Available: https://easychair.org/publications/paper/wFh9

[31] H. Beglerovic, M. Stolz, and M. Horn, "Testing of autonomous vehicles using surrogate models and stochastic optimization," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[32] A. Gambi, W. Hummer, and S. Dustdar, "Testing elastic systems with surrogate models," in *Proceedings of the 1st International Workshop on Combining Modelling and Search-Based Software Engineering*, ser. CMSBSE '13. IEEE Press, 2013, pp. 8–11.

[33] M. Biagiola and P. Tonella, "Testing of deep reinforcement learning agents with surrogate models," *ACM Trans. Softw. Eng. Methodol.*, nov 2023. [Online]. Available: https://doi.org/10.1145/3631970

[34] T. Laurent, P. Arcaini, F. Ishikawa, H. Kawamoto, K. Sawai, and E. Muramoto, "Investigating multi- and many-objective search for stability-aware configuration of an autonomous delivery system," in *2023 30th Asia-Pacific Software Engineering Conference (APSEC)*, Dec 2023.

[35] T. Laurent, P. Arcaini, X. Zhang, and F. Ishikawa, "Metamorphic testing of an autonomous delivery robots scheduler," in *2024 IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2024.