ZĂVOIANU, A.-C., SAMINGER-PLATZ, S., ENTNER, D., PRANTE, T., HELLWIG, M., SCHWARZ, M. and FINK, K. 2018. Multi-objective optimal design of obstacle-avoiding two-dimensional Steiner trees with application to ascent assembly engineering. *Journal of mechanical design* [online], 140(6), article number 061401. Available from: https://doi.org/10.1115/1.4039009

Multi-objective optimal design of obstacle-avoiding two-dimensional Steiner trees with application to ascent assembly engineering.

ZĂVOIANU, A.-C., SAMINGER-PLATZ, S., ENTNER, D., PRANTE, T., HELLWIG, M., SCHWARZ, M. and FINK, K.

2018

© ASME

OpenAIR @RGU

This document was downloaded from https://openair.rgu.ac.uk

CC BY

American Society of Mechanical Engineers

ASME Accepted Manuscript Repository

Institutional Repository Cover Sheet

First Last

ASME Paper Title: Multi-objective optimal design of obstacle-avoiding two-dimensional Steiner trees with application to ascent assembly engineering.

Authors: ZĂVOIANU, A.-C., SAMINGER-PLATZ, S., ENTNER, D., PRANTE, T., HELLWIG, M., SCHWARZ, M. and FINK, K.

ASME Journal Title: Journal of Mechanical Design

Volume/Issue 140(6) Date of Publication (VOR* Online) 26.03.2018

ASME Digital Collection URL: https://asmedigitalcollection.asme.org/mechanicaldesign/article-abstract/140/6/061401/376352

DOI: https://doi.org/10.1115/1.4039009

*VOR (version of record)

Multi-Objective Optimal Design of Obstacle-Avoiding 2D Steiner Trees with Application to Ascent Assembly Engineering

Alexandru-Ciprian Zăvoianu*
Susanne Saminger-Platz
Department of Knowledge-Based Mathematical Systems
Johannes Kepler University Linz
Altenbergerstraße 69, 4040 Linz, Austria
Email: {Ciprian.Zavoianu, Susanne.Saminger-Platz}@jku.at

Doris Entner
Thorsten Prante
Design Automation
V-Research GmbH - Industrial Research and Development
Stadtstraße 33, 6850 Dornbirn, Austria
Email: {Doris.Entner, Thorsten.Prante}@v-research.at

Michael Hellwig
Research Centre Process and Product Engineering
Vorarlberg University of Applied Sciences
Hochschulstraße 1, 6850 Dornbirn, Austria
Email: Michael.Hellwig@fhv.at

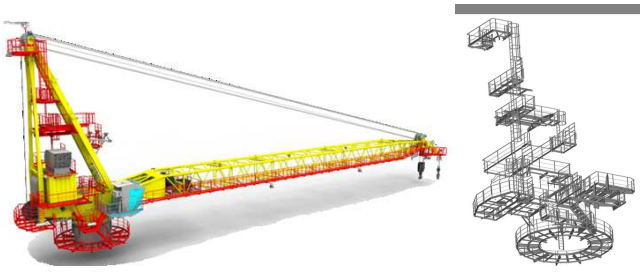
Martin Schwarz
Klara Fink
Technology Management
Liebherr-Werk Nenzing GmbH
Dr.-Hans-Liebherr-Str. 1, 6710 Nenzing, Austria
{Martin.Schwarz, Klara.Fink}@liebherr.com

We present an effective optimization strategy that is capable of discovering high-quality cost-optimal solution for 2D path network layouts (i.e., groups of obstacle-avoiding Euclidean Steiner trees) that, among other applications, can serve as templates for complete ascent assembly structures. The main innovative aspect of our approach is that our aim is not restricted to simply synthesizing optimal assembly designs with regard to a given goal, but we also strive to discover the best trade-offs between geometric and domain-dependent optimal designs. As such, the proposed approach is centered on a variably constrained multi-objective formulation of the optimal design task and on an efficient coevolutionary solver. The results we obtained on both artificial problems and realistic design scenarios based on an industrial test case empirically support the value of our contribution to the fields of optimal obstacle-avoiding path generation in particular and design automation in general.

1 Introduction

This work is primarily motivated by theoretical and practical considerations related to engineering design automation processes. Concretely, the presented approach stems from the desire to automatically generate cost-optimal *complete ascent assembly structures* (CAA-Structures) – external access structures required by cranes, building facades, over-sized industrial machines, etc. Fig. 1 shows an example of a fairly complex CAA-Structure that is itself composed from several types of ascent assembly modules (i.e., sub-assemblies) like rectangular and round platforms, stairs placed at different inclinations, and ladders.

*Address all correspondence related to this article to this author.



(a) (b)

Fig. 1. An example of an offshore crane where different ascent assembly modules highlighted in red (a) are combined to form a fairly complex CAA-Structure (b)

The task of designing individual ascent assembly modules, although very important, is rather repetitive and time consuming. Recently, in light of strong financial and operational incentives, there has been a consistent and successful effort to standardize individual ascent assembly modules and to automate their design process [1]. As a result, the task of automating the design of cost-optimal CAA-Structures has itself become a feasible undertaking since it can be regarded as a search for a 3D "skeleton" (i.e., a pathway) that indicates which ascent assembly modules are required and how they should be placed in ordered to obtain a CAA-Structure that provides the desired level of access with minimal costs[1]. In other words, discovering cost-optimal CAA-Structures can be addressed as an automated design synthesis challenge [2–4].

Within the broader goal of obtaining cost-optimal CAA-Structures [5], the main novelty of our approach lies in its ability to discover an accurate image of the trade-offs between standard CAA-Structures (that must obey certain constraints regarding existing ascent assembly modules and ways of combining them) and more creative CAA-Structure designs that can be obtained when freely exploring the design space. Thus, our idea was to develop an efficient optimization approach centered around a multi-objective problem formulation that can ultimately offer human designers / decision makers (DMs) insight regarding the particular challenges of the CAA-Structure design task at hand as well as tentative solutions with various degrees of "conformity" to standard design practices. Apart from the focus on optimal trade-off identification, the present work also extends and improves on the preliminary study in [5] by:

1. examining a more advanced variation of the optimization problem that also considers disjoint CAA-Structures;
2. analyzing and evaluating optimization performance more in depth on a larger and more varied set of benchmark and industrial test cases.

It is noteworthy that by tackling the design synthesis task using the proposed optimal trade-off search, we intrinsically consider domain-based design restrictions (best practices) as *soft constraints* and, during the search, we automatically single-out those minor violations (exceptions to the domain-based design rules) that can deliver a maximal reward (i.e., reduction of costs). Therefore, from an application point of view, our approach is also innovative as, at the end of the optimization process, it is able to provide the DM with both domain-optimal CAA-Structure designs as well as optimal variations of these designs that can considerably reduce costs.

The remainder of this work is organized as follows: Section 2 is dedicated to formalizing the CAA-Structure optimization task based on the chosen modeling procedure and on highlighting its connection with the well-known and widely encountered (obstacle-avoiding) minimum Steiner tree problem; Section 3 presents the motivation for and the in-depth description of the proposed optimization strategy; Section 4 describes the setup of the numerical (optimization) experiments as well as the benchmark and industrial test cases; Section 5 contains discussions of the results while the conclusions and outlook are covered in Section 6.

2 Modelling and formal problem statement

In spite of the apparent simplicity suggested by the need to discover a 3D "skeleton", there is still a large set of particularities and uncertainties associated with real-life CAA-Structure design tasks in modern engineer-to-order environments. In order to have a relevant but accessible formulation for analyzing and comparing the performance of various proofs of concept and optimization algorithms, for this initial research stage, we introduce and operate with a 2D model abstraction of the optimal design task [5].

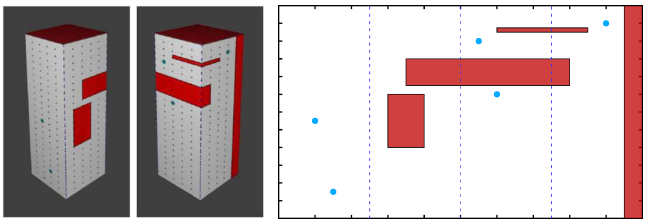
2.1 Description of model abstraction

A user that wishes to generate a cost-optimal CAA-Structure is expected to provide at least three inputs: a hull of a 3D model of the solid base (i.e., support object) to which the CAA-Structure is to be attached, a set of desired points of access on this structure and information regarding potential obstacles that are defined on the 3D solid base object. The latter requirement is extremely relevant as obstacles are meant to indicate severe restrictions (i.e., *hard constraints*) regarding the placement of ascent assembly modules in certain areas.

For example, in Fig. 2a, we illustrate a simplified design case that involves a cuboid structure, five access points and four obstacle areas that are spread across three faces of the cuboid. A far clearer representation of this academic automated design scenario can be obtained by unfolding the 3D model. The result of the unfolding procedure, shown in Fig. 2b, is a 2D design surface that is characterized by a left edge - right edge continuity – i.e., line segments exiting the left edge at a certain height and orientation, should enter the right edge at the same height and orientation in order to model the circular structure of the facade. More importantly, as a result of the unfolding, the task of discovering a cost-optimal 3D

[1] We consider a general cost function that may or may not include a monetary component. Conversely a maximization problem could also be formulated when using a general utility function.

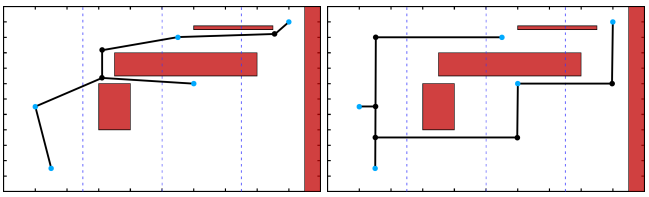
"skeleton" of the ascent assembly is transformed into that of synthesizing a simpler 2D design "skeleton": a *cost-optimal 2D path network layout* on the 2D design surface that links all the points of interest while avoiding the obstacle areas.



(a) (b)

Fig. 2. A 3D model and the corresponding 2D design plane obtained after unfolding. Access points are marked with blue circles and obstacles are marked with red.

In Fig. 3 we present two different 2D path network layouts that link the definition points illustrated in the academic example from Fig. 2. It is noteworthy that both path network layouts make use of several auxiliary points (3 for the design in Fig. 3a and 5 for the design in Fig. 3b) and that the number and the placement of these auxiliary points uniquely determines the design of the pathway. Another observation is that while both path network layouts are valid, as they don't span through obstacle regions, neither are optimal as their overall length is not minimal. Furthermore, within the context of designing CAA-Structures, the 2D "skeleton" from Fig. 3b might be considered more realistic as it only uses horizontally and vertically placed sub-assemblies (i.e., platforms and ladders), while the solution from Fig. 3a achieves a smaller overall total path length by placing its contained sub-assemblies at non-standard angles.



(a) (b)

Fig. 3. Example of 2D path network layouts that link (blue) definition points while avoiding (red) obstacle areas

The main advantage of the unfolding approach consists in its ability to generate valuable proof-of-concept optimization test scenarios that are both simple and useful (i.e., applicable to real-life CAA-Structure design synthesis tasks, as shown in Section 4.3). The main disadvantage lies in the restrictiveness of the 2D abstraction, as very complicated 3D models are very hard or even impossible to unfold (e.g., the offshore crane presented in Fig. 1a).

Taking into consideration the main purpose of a CAA-Structure is to provide access to various parts of a base object, it makes sense to classify the above mentioned *access points* into:

- *entry points* that are usually placed (at the bottom of the base object) in such a way as to facilitate human access on the CAA-Structure
- *work points* that are placed (at the upper levels of the base object) in areas that must be made accessible to humans for operational and maintenance purposes

The above classification of access points is quite important because it helps differentiate between two main types of 2D path network layouts:

1. *Fully connected layouts* are the expected solutions of design scenarios that contain a single entry point or of design scenarios that impose a secondary restriction that every access point should be reachable from any other access point.
2. *Disjoint layouts* that are made up of several smaller fully connected sub-layouts (one for each entry point) that contain among themselves all the work points defined in the design scenario.

While disjoint layouts are likely design solutions in practice, for the sake of brevity, clearness, and generality, in Section 2.2 we first recap the cost-optimization problem formulation for fully connected layouts [5]. The more challenging disjoint formulation is obtained using a minimal modification, but both types of 2D path network layout problems are subsequently solved using the same multi-objective based optimization strategy in order to demonstrate its robustness (i.e., performance invariance with regard to number and order of access and entry points).

In Section 2.2 we provide formal arguments that the resulting *2D path network layout problem* is by no means trivial as solving it in its simplest form actually means discovering the *minimum Euclidean Steiner tree* that covers the definition points. This means that the multi-objective based strategy for designing optimal pathways we describe in the present article is suitable for several other application domains where the generation of minimum Euclidean Steiner trees is a key interest: integrated circuit design [6, 7], transportation systems [8], computer networks [9], urban planning [10], unmanned aerial vehicle routing [11]. Literature on obstacle-avoiding minimum Steiner trees is mainly focused on the rectilinear case [12, 13] as this version has direct application to the efficient design of electrical circuits. Nevertheless, a few approximation methods for the (non-rectilinear) Euclidean version have also been proposed [14, 15]. When comparing with all other approaches, the efficient and generic obstacle-avoiding Steiner tree synthesis strategy we presently propose has two important advantages that should make it of interest to a wider public:

1. It can generate a wide range of Pareto-optimal obstacle-avoiding Steiner trees – e.g., ranging from rectilinear to Euclidean as shown in Fig. 11 – for a given pathway design problem during a single run as it can account for multiple design objectives during the search.

2. It can optimally decompose a cost optimal design into a specified number of independent obstacle-avoiding Steiner trees (e.g., Figs. 6e and 10d).

2.2 Formalization of the path network layout problem

When considering a set of n user defined *access points* / definition vertices $P = \{p_1, \ldots, p_n\}$, the goal of the (fully connected) 2D optimal path network layout problem is to discover a (graph) structure T of minimal cost that links all these points. T must obviously span all the access points but it may also contain up to k well-placed extra points (2D vertices) $S = \{s_1, \ldots, s_k\}$ that help minimize the total cost of T. Thus, E, the set of possible edges that contains all the segments that can be used to construct T, is defined over the union $P \cup S = \{p_1, \ldots, p_n, s_1, \ldots, s_k\}$. When considering a positive cost for connecting any two points, it is obvious that T is in fact a tree. Formally, the resulting minimal path optimization task can be defined as: *"Determine* $k \in \mathbb{N}$ *and* $s_1, \ldots, s_k \in \mathbb{R} \times \mathbb{R}$ *in order to minimize*

$$f_1(p_1, \ldots, p_n, s_1, \ldots, s_k) = \sum_{(ij) \in E} c(i,j) x_{(ij)}, \tag{1a}$$

subject to:

$$x_{(ij)} \in \{0,1\},\ \forall (ij) \in E \ \ and \tag{1b}$$

$$\sum_{(ij) \in E} x_{ij} = (n+k) - 1 \ \ and \tag{1c}$$

$$\sum_{(ij) \in E, i \in F, j \in F} x_{ij} \leq |F| - 1,\ \forall F \subseteq P \cup S, \tag{1d}$$

where $G = (P \cup S, E)$ *is a complete graph."* The constraint from Eqn. (1b) simply enforces clarity: a certain edge either is or is not part of T. The constraint from Eqn. (1c) ensures that T is fully connected and Eqn. (1d) guarantees that no cycles can be formed (and thus T is a tree).

In order to obtain the problem formulation for disjoint layouts, let us consider a set $P_E \subseteq P, P_E \neq \emptyset$ that reunites the (user-defined) *entry points*. Constraint (1c) must be adapted to $\sum_{(ij) \in E} x_{ij} = (n+k) - |P_E|$ and we must additionally ensure that each disjoint component contains an entry point, i.e., $\forall p^* \in P \setminus P^E, \exists L \subseteq P \cup S : p^* \in L \wedge L \cap P^E \neq \emptyset$ such that:

$$\sum_{(ij) \in E, i \in L, j \in L} x_{ij} = |L| - 1. \tag{1e}$$

The function $c(i,j)$ from Eqn. (1a) denotes the cost of linking vertices i and j. In the case of ascent assemblies, this cost usually has a strong financial nature and can be defined as the combined price of individual modules (i.e., platform, stair, and ladder segments) required to construct a walkway between points i and j and of connecting these modules (e.g., welding). While it is expected that, in the general case, $c(i,j)$ is proportional to the Euclidean distance between the two vertices, in more realistic scenarios, obstacles and other penalties do influence the cost function.

For example, when considering a slightly more realistic description of optimal layouts for CAA-Structures, one would likely consider inside $c(i,j)$ a large penalty $\Gamma_{(ij)}$ for assembly modules that extend into obstacle areas when connecting vertices i and j and another smaller penalty for assembly modules that are not placed at a preset angle requirement – e.g., platforms should be placed at an angle of exactly 0° to the horizontal axis, stairs at 45°, and ladders at 90°. All these "allowed/preferred" design angles should be provided as a user defined set, e.g., $U = \{0, 45, 90\}$. The resulting angle-aware cost function could be defined as:

$$c(i,j) = dist(i,j) \left(1 + \frac{\min B_{(ij)}}{100} z\right) + \Gamma_{(ij)} \tag{2}$$

where z is a parameter ($0 \leq z \leq 4$) that controls the magnitude of the angle penalty and $B_{(ij)}$ is a set that contains the absolute differences between $\alpha_{(ij)}$ – the horizontal angle of the segment (ij) – and the allowed placement angles stored in U. For instance, given the aforementioned composition of U , $B_{(ij)} = \{|\alpha_{(ij)} - 0|, |\alpha_{(ij)} - 45|, |\alpha_{(ij)} - 90|\}$. For the set of tests we report over in the present study, $dist(i,j)$ marks the 2D Euclidean distance between vertices i and j.

The *variably constrained* cost-template from Eqn. (2) provides a simple but effective way to introduce via predefined parameters (like U and z) domain-specific constraints in the optimal design formulation for 2D path network layouts.

It is noteworthy to remark that when $z = 0$ in Eqn. (2) and one does not consider obstacle areas and a left-right continuity of the design plane, $c(i,j)$ is reduced to the Euclidean distance and Eqns. (1a) (1b), (1c), and (1d) form the definition of the well-known *Euclidean Steiner Tree Problem* (ESTP) [16]. Although they represent the simplest cases of the (fully connected) 2D path network layout problems we aim to solve, ESTPs are proven to be NP-hard [17]. Nevertheless, ESTPs have also been intensively studied by mathematicians and computer scientists and this opens up the possibility to compare (in part) our proposed solving strategy with other results from literature on standard benchmarks. In the context of ESTPs, the k points that help minimize the 2D path layout between the access points are called *Steiner points* and throughout this work we shall also maintain this naming in the context of optimal path network layouts for CAA-Structures.

Finally, in the context of path network optimization tasks for ascent assemblies, opting for a value of $z = 0$ in Eqn. (2) would result in a problem definition that enables the optimizer to freely explore the design space and quite possibly discover innovative designs (i.e., innovative ways of connecting the desired access points). However, the best results of such an "open" definition would (likely) only be interpreted as optimal design "suggestions" as building them to specification would be unfeasible. When opting for a larger value of the penalty parameter z and a realistic list of stan-

dard allowed angles, good results of the more "restricted" path optimization problem are far more likely to resemble "blueprints" of the ascent assembly.

Our idea is to solve both the "open design" and "restricted design" formulations at the same time (even if they are conflicting), by combining them inside a multi-objective optimization problem (MOOP). The advantage of pursuing such an approach is directly linked to what one commonly expects when solving a MOOP. Whenever multiple objectives are involved, there is usually no single design that solves the problem as there is no single design that is superior across all objectives. The solution to a MOOP [18] usually consists of a *set of Pareto optimal designs* (notation: PS for Pareto set). Each element from the PS has the property that it is not fully dominated (i.e., worse or equal with respect to all objectives) by any other possible element in the design space. Since Pareto optimal sets can be infinite and / or very hard to determine, one is usually satisfied with finding a reasonably accurate (discrete) approximation in the form of a Pareto non-dominated set (notation: PN) which is based on a weaker membership condition: an element of the PN cannot be fully dominated by any other element of the PN. This means that the PN captures the trade-offs between the objectives to be optimized. In our case, the PNs will provide a decision maker with a set of optimal designs that illustrate the trade-offs of moving from design "suggestions" (geometric optimality) to design "blueprints" (domain specific optimality). The final decision regarding the best individual design for a given problem is in essence a subjective one, but having an objective image of the existing trade-offs should enable the DM to make a more informed choice.

3 Optimization procedure

The NP-hard nature of ESTPs – the easiest type of fully connected 2D path network layout problems we aim to solve – motivates our strong preference for a meta-heuristic-based solver. Furthermore, since our goal is to have a multi-objective approach, the evolutionary-based optimization paradigm represented an obvious choice as multi-objective evolutionary algorithms (MOEAs) have been (historically) proven to be the most successful meta-heuristic solving technique [19] with various useful applications to engineering design [20–23]. In light of this, some of the lexicon throughout the remainder of this work is tailored for the field of evolutionary computation.

It must be emphasized that, despite their proven robustness, much like natural evolution itself, MOEAs are stochastic processes for which there are no general guarantees with regard to global solution optimality, success ratio and time required for convergence. Thus, like all meta-heuristic optimization strategies, MOEAs are generally intended as a measure of last resort for scenarios in which deterministic solvers perform rather poorly (and most NP-hard problems offer prime examples of such scenarios).

3.1 Solution codification

A very important aspect of trying to solve the problem described in Section 2.2 is represented by the encoding of individuals / candidate solutions. First and foremost, a good encoding should be simple (*general*) in order to be compatible with many fitness assessment strategies and in order to allow for an immediate extension to 3D scenarios. Secondly, the encoding should also be *flexible* as the number of Steiner points required by each problem is unknown. Although the latter characteristic seems to hint towards a variable-length encoding, we argue in favour of a fixed-length variant in which the maximal number of the Steiner points expected to be discovered (i.e., k^*) is preset at a sufficiently large level. For example:

- in the case of ESTPs, one can use the mathematically proven [16] upper bound $k^* = n-2$
- in the case of all the ascent assembly optimization problems presented in Section 4 we experimented with several settings in the range $n \leq k^* \leq 3n$.

The task of "deciding" the exact number of Steiner points required for solving the problem at hand is "passed" to the fitness assessment method described in the next section. Apart from the extra simplicity that enables the usage of various standard genetic operators, our choice for a fixed-length encoding is also motivated by the desire to counteract potential solution bloating - a well-known and harmful phenomenon in terms of both solution quality and convergence speed that is associated in the field of evolutionary computation (genetic programming in particular) with combinations of strong (evolutionary) selection pressure and variable-length encodings [24].

After opting for fixed-length encodings, we adopted a basic real-valued vector representation $\vec{x} = (x_1, x_2, \ldots, x_{2k^*-1}, x_{2k^*})$ of solution candidates. The understanding is that, given the 2D encoded vertex $v_{(i,\vec{x})}$ $1 \leq i \leq k^*$:

- x_{2i-1} denotes the horizontal coordinate of $v_{(i,\vec{x})}$.
- x_{2i} denotes the vertical coordinate of $v_{(i,\vec{x})}$.

A visual example of the proposed encoding is presented in Fig. 4.

The minimum and maximum ranges for $x_i \in \vec{x}$ are set according to the design scenario definition limits in the case of CAA-Structures and to the extreme coordinate values of the definition points in the case of ESTPs. By definition, Steiner points cannot exist outside these minimum and maximum ranges.

3.2 Fitness assessment of primary objectives

Let $o_1(\vec{x})$ denote the ability of the vertices encoded in a given candidate solution $\vec{x}$ to minimize Eqn. (1a) when considering a Euclidean cost function. In order to estimate $o_1(\vec{x})$, we employ a two-step process:

- Firstly, we build the union between all the k^* vertices encoded by $\vec{x}$ and the n definition points of the optimization scenario: $S_{\vec{x}} = \{v_{(1,\vec{x})}, \ldots, v_{(k^*,\vec{x})}\} \cup \{p_1, \ldots, p_n\}$.

- Secondly, starting with p_1, we apply Prim's algorithm [25] in order to construct $MT_{n,\vec{x}}$ – the *partial minimum spanning tree* (MST) over the set $S_{\vec{x}}$ that contains all n definition points. $MT_{n,\vec{x}}$ is a partial MST because the construction / synthesis process is interrupted once all the definition points have been added to the tree.

Any vertex encoded in $\vec{x}$ that remains unlinked by $MT_{n,\vec{x}}$ has the property that its placement is highly likely not to improve in any way the formation of an optimal-cost path between all the definition points $\{p_1,\ldots,p_n\}$ – i.e., this vertex is deemed as having a low chance of being a potential Steiner point. We mark with $s_{i,\vec{x}}$, $i \in \{1,\ldots,m\}$, $m \leq k^*$ the vertices encoded in $\vec{x}$ that are part of $MT_{n,\vec{x}}$. Compared with the unlinked vertices, any $s_{i,\vec{x}}$ has a better chance of being useful in constructing an optimal path between the definition points and is thus deemed a *potential Steiner point*. When considering previous notations, and denoting with $\Phi(MT_{n,\vec{x}})$ the total cost associated with $MT_{n,\vec{x}}$, we have that $\Phi(MT_{n,\vec{x}}) = f_1(p_1,\ldots,p_n,s_{1,\vec{x}},\ldots s_{m,\vec{x}})$ where function f_1 is defined in Eqn. (1a).

We argue that $o_1(\vec{x})$ can be well approximated by $\Phi(MT_{n,\vec{x}})$ because the closer the set of potential Steiner points in $\vec{x}$ is to $\{s_1,\ldots,s_k\}$, i.e., to the actual Steiner point set that represents the solution to Eqn. (1a), the closer $f_1(p_1,\ldots,p_n,s_{i,\vec{x}},\ldots s_{m,\vec{x}})$ is to $f_1(p_1,\ldots,p_n,s_1,\ldots s_k)$.

In order to better illustrate how fitness is estimated when considering the fixed-length real-valued encoding, in Fig. 4 we plot the partial MST of a candidate solution $\vec{x}$ that contains 10 design variable as it encodes a total of $k^* = 5$ 2D vertices. We assume that $m = 3$ and consider a hypothetical mapping of the three potential Steiner points (black) and of the two unlinked points (grey). Regarding the two unlinked points, it is important to note that encoded vertex no. 3 – defined by x_5 and x_6 – is placed inside an obstacle area (so linking it would incur a severe penalty) while encoded vertex no. 5 – defined by x_9 and x_{10} – is placed at a considerable distance from any other point. In other words, a very good solution candidate must have:

- sufficient potential Steiner points placed in nearly ideal locations;
- all the other encoded vertices placed out of the way, in positions that do not negatively influence the construction of $MT_{n,\vec{x}}$.

It is noteworthy that in the case of fully connected 2D path network layouts, the construction process for $MT_{n,\vec{x}}$ can be started by automatically initializing the tree structure with any definition point (not just p_1). In the case of disjoint layouts, the only difference is that the construction process must be initialized by adding all the user-defined *entry points*.

Furthermore, the computation of $MT_{n,\vec{x}}$ is compatible with various settings of the cost function. For instance, in order to define a multi-objective optimization problem that aims to present the DM with the trade-offs between geometrical (i.e., Euclidean) and domain specific optimal solutions, apart from $o_1(\vec{x})$, one should also define a second primary objective $o_2(\vec{x})$ by parameterizing the variably constrained

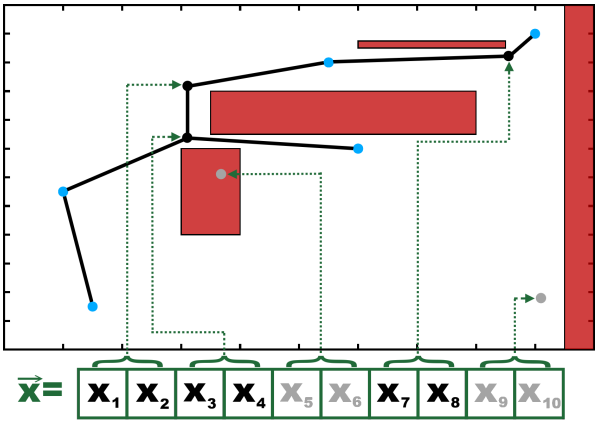


Fig. 4. Example of a solution candidate $\vec{x}$ for which $k^* = 5$ and $m = 3$ and of the resulting $MT_{n,\vec{x}}$

cost-template from Eqn. (2) with relevant angle-restrictions. The compliance of a candidate solution with $o_2(\vec{x})$ can be estimated by simply using the angle-constrained cost function during the computation of the partial MST. Extensions of the $MT_{n,\vec{x}}$-based fitness assessment strategy to 3D cases are straightforward.

In light of all the aforementioned reasons, one can argue that, despite its apparent complexity, the proposed fitness assessment procedure is very well suited for the considered optimization problem.

3.3 Artificial secondary objective

When considering the main motivation behind the present work (i.e., optimizing real-life industrial designs), it is highly likely that a more advanced future model abstraction might yield *secondary requirements regarding optimality*. For instance, these requirements might relate to:

- the complexity of the overall CAA-Structure design (i.e., number of different module types that is required),
- ensuring different levels of ease-of-access for different definition points,
- CAA-Structure building time given present stocks of individual ascent assembly modules,
- concerns regarding aesthetic impact
- theoretical considerations regarding algorithm performance.

Such possible secondary requirements appear to be rather conflicting with the currently identified primary one (i.e., geometrically-related cost minimization). Furthermore, modeling some of these secondary requirements by extending the cost function in Eqn. 2 via penalties and rewards is expected to be extremely cumbersome. Alternatively, formalizing them as optimization objectives in their own right would be more natural and should yield better results from the perspective of a DM.

Motivated largely by the previous considerations but also by initial attempts to simultaneously optimize $o_1(\vec{x})$ and $o_2(\vec{x})$ that were less successful than anticipated (showing signs of premature convergence), we defined an artificial sec-

ondary objective $o_a(\vec{x})$. This secondary objective is designed to be (slightly) conflicting with the main ones described in Section 3.2 and is defined as:

$$o_a(\vec{x}) = \frac{1}{n-1}\left(\sum_{r=2}^{n} impr_{MST(r)} * 2^{size_{MST(r)}}\right) - 1.1^m, \quad (3)$$

where:

$impr_{MST(r)} = \Phi(MT_{r,\vec{x}}) - \Phi(MST_r)$ *and*

$size_{MST(r)} = 3 - \frac{3\Phi(MST_r)}{\Phi(MST_n)}$.

Inside Eqn. (3), when considering that p_r is the r^{th} user defined access point, in an analogous way to $\Phi(MT_{n,\vec{x}})$, we have that:

- $\Phi(MST_r)$ is the cost of the minimum spanning tree constructed over the set $\{p_1,\ldots,p_r\}$, i.e., $\Phi(MST_r) = f_1(p_1,\ldots,p_r)$;
- $\Phi(MT_{r,\vec{x}})$ is the total cost of the *partial minimal spanning tree* constructed over the union $S_{\vec{x}} = \{v_{(1,\vec{x})},\ldots,v_{(k^*,\vec{x})}\} \cup \{p_1,\ldots,p_r\}$

This means that $o_a(\vec{x})$ computes the average level to which $\vec{x}$ is able to solve Eqn. (1a) for every incremental subset of definition points that is obtained when constructing the minimal spanning tree over $\{p_1,\ldots,p_n\}$. The smaller the subset the more important it is weighted inside the average and there is a small bonus for candidate solutions that achieve good results with a reduced number m of potential Steiner points.

Finally, we have chosen to always add $o_a(\vec{x})$ to the 2D path network optimization problems to be solved, as an extra objective to be minimized. This means that:

- when wishing to simultaneously optimize both Euclidean and domain specific cost functions, we consider a MOOP with three objectives (the two primary ones and $o_a(\vec{x})$).
- even when the overall goal is to optimize a single cost function (e.g., ESTPs), we formulate a MOOP with two objectives (by also considering $o_a(\vec{x})$ alongside the primary objective).

Although empirically validated by all the results presented in Section 5, the inclusion of $o_a(\vec{x})$ alongside the main path minimization objective(s) is highly counter-intuitive. The reasoning for this decision is two-fold:

- $o_a(\vec{x})$ is engineered to induce both some level of niching during the evolutionary search as well as a biasing of the multi-objective search towards robust candidate solutions that encode potential Steiner points which are generically well placed – i.e., able to improve the total minimal path in key locations that are common to many sub-paths.
- Having a (more complex) multi-objective formulation enables us to check whether our assumption that $\Phi(MT_{n,\vec{x}})$ is a good enough approximation for $o_1(\vec{x})$ and / or $o_2(\vec{x})$ also holds when faced with a conflicting optimization objective that, to a certain extent, aims to steer the evolutionary search towards 2D path layouts that are not necessarily cost-optimal.

3.4 The Multi-Objective Solver

In order to solve the previously introduced multi-objective optimization problems, we performed a limited set of initial tests with NSGA-II [26] – a well-known classical multi-objective evolutionary algorithm – and with DECMO2 [27]. The latter is a newer hybrid and adaptive evolutionary approach specially designed for rapid convergence on a wide class of problems. DECMO2 was designed to capitalize on previous insights [28] that very competitive results can be obtained by a *cooperative coevolutionary strategy* that combines the SPEA2 evolutionary model [29] that uses simulated binary crossover [30] and polynomial mutation operators [31] with the GDE3 [32] / DEMO [33] evolutionary model that exploits the very good performance exhibited by the *DE/rand/1/bin* differential evolution operator [34] on continuous optimization problems.

The third MOO paradigm incorporated in DECMO2 comes in the form of an archive of elite solutions that is maintained according to a decomposition-based principle similar to the one proposed in MOGLS [35] and popularized by MOEA/D-DE [36]. This role of this archive is to enable DECMO2 to deliver well-spaced Pareto non-dominated sets.

As DECMO2 exhibited a better overall balance between convergence speed and final solution quality (in terms of both Pareto optimality and PN spacing), we adopted it as the default solver for all the numerical optimization tests we report over.

4 Experimental setup

Although the performed numerical experiments range from simple (but well-studied) ESTP-related optimization tasks to ever more complicated / realistic 2D path network layout that have immediate applicability in optimal design of CAA-Structures, we have chosen to use a fixed and largely standard parameterization of DECMO2 in order to also evaluate the robustness of the multi-objective solver.

4.1 DECMO2 parameterization

For all the numerical experiments we report on, we used a total population size of a_{size} =400 for DECMO2 and the standard (literature recommended) parameter settings for the coevolved subpopulations of the solver.

In the case of MOOPs with two objectives (one primary + the artificial) we evaluated 100.000 solution candidates during each optimization run (i.e., 250 generations) and in the case of MOOPs with three objectives (two primary + the artificial) we allowed DECMO2 to generate 400.000 solution candidates. We report on the best result out of 3 independent repeats for each numerical experiment.

Since the artificial objective of our MOOPs (described in Section 3.3) is mainly used to ensure spacing in the MOEA during the run and does not have any practical importance for DMs when assessing the overall result of the optimization

runs, we always only report the best discovered solution(s) with regard to the primary objectives defined in Section 3.2. This means that for MOOPs with two objectives (one primary + the artificial) we only report the best discovered design with regard to either $o_1(\vec{x})$ (Euclidean costs) or $o_2(\vec{x})$ (domain-specific costs that reflect angle-wise penalties).

4.2 ESTP benchmark and academic test cases

In order to demonstrate the ability of our approach, we compared the results obtained by DECMO2 on 25 problems from a benchmark ESTP set [37] with those of two reference solvers: one based on a geometrically motivated heuristic [38] and another one that is based on artificial neural networks [39]. For comparative purposes we also refer the best known results suggested in [37], although the means through which these results were determined is not described.

For initial insight on how our method performs on optimization scenarios that are more representative for the ascent assembly domain, we proceeded to apply DECMO2 on 6 academic test cases. The first one (notation: A1) is illustrated at the beginning of this paper in Fig. 2. The other five academic test cases (notation: A2 through A6) are derivations based on the challenging placement of the 14 access points from problem no. 12 of the ESTP benchmark set. Their aim is to test if and how DECMO2 is able to adapt optimal designs when confronted with an ever more difficult placement of obstacles and multiple entry points.

On all the ESTP and academic tests we used the setting $z = 0$ to parameterize the cost function from Eqn. (2) and thus optimized the minimum Euclidean distance of the fully connected 2D path network layout.

4.3 Industrial test case

The most realistic cost-optimal CAA-Structure design scenario we investigated was proposed by Liebherr-Werk Nenzing GmbH (LWN) [40] – a manufacturer of a wide range of products including various types of cranes. More specifically, we investigated cost-optimal CAA-Structures that allow access to user-specified regions of interest on the gantry of a mobile harbour crane (please see Fig. 5a). Figure 5b presents an expert-designed CAA-Structure attached to the gantry and we aim to use the unfolding-based 2D model abstraction from Fig. 5c (obtained by vertically stacking two cuboids) to explore complementary optimal designs that might provide interesting insights to LWN.

Across all the industrial optimization scenarios we used four different cost settings (i.e., primary objectives):

- C1 — The first cost setting uses a value of $z = 0$ to parameterize the cost function from Eqn. (2). Given the infinite degrees of freedom, optimal designs discovered for this setting are expected to have the smallest total path network (Euclidean) distance and can be used as a generic structural reference when assessing more constrained cost-optimal designs.
- C2 — The second cost setting uses a value of $z = 4$ and a list of preferred design angles $U = \{0, 45, 90\}$ and aims to deliver ascent assembly designs that only use the three standard assembly components: horizontal platforms, stairs and vertical ladders.
- C3 — The third cost setting uses a softer angle-wise constraint factor of $z = 1$ and a minimal list of preferred design angles $U = \{0, 90\}$ and aims to deliver ascent assembly designs that have a real-life minimal cost as they only use horizontal platforms and vertical ladders.
- C4 — The fourth cost setting uses a value of $z = 4$ and a minimal list of preferred design angles $U = \{0, 30, 45\}$ and aims to deliver CAA-Structures that offer a higher degree of access (no mandatory use of hands) by only using platform modules and two types of staircase modules (mild and regular inclination).

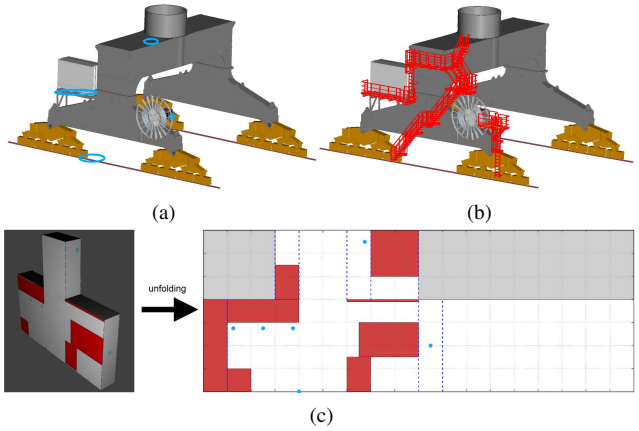


Fig. 5. A CAD model of the gantry of a Liebherr mobile harbour crane with highlighted user-specified access points (a) expert-designed ascent assembly solution (b) and complementary 3D and unfolded 2D model abstraction (c)

It is important to observe that, when reporting to the definitions of the primary objectives from Section 3.2, C1 maps perfectly to $o_1(\vec{x})$ as it aims to minimize the Euclidean distance while C2, C3, and C4 are different variants of $o_2(\vec{x})$ that enforce different domain specific angle preferences.

Firstly, in order to obtain an image of the best achievable Euclidean and domain-specific cost-optimal designs of the LWN CAA-Structure, we considered three test case variations involving a single primary objective (notation: TC1, TC2, and TC3) in which a single ground access (i.e., entry) point is placed at different positions along the horizontal axis.

Secondly, we also analyzed a fourth scenario (notation: TC4) that, while still considering a single primary objective during the optimization, explores the expert approach of using two ground entry points. The aim of TC4 is to explore the reduction in cost that can be achieved by investigating disjoint layouts of the CAA-Structure.

Thirdly, the most complex optimization we carried out based on the industrial scenario provided by LWN aimed to discover the best trade-offs in the context of TC1 between optimal solutions under the Euclidean C1 cost setting and optimal solutions under a stricter (i.e., $z = 4$) domain specific

C3 cost setting.

5 Results

While the optimal design results related to the industrial CAA-Structure are expected to be more interesting to the reader, we would like to draw attention that the performance of our approach on various ESTPs and academic test cases is also very important as it offers good insight into the efficiency and generality of the presently proposed method of designing cost-optimal 2D path network layouts.

5.1 Performance on artificial problems

The comparative performance of our approach on benchmark ESTPs is presented in Tab. 1 and indicates that our solving strategy based on DECMO2 and a MOOP formulation (with one primary objective and one artificial objective) is very competitive for ESTP instances that have a low-to-medium number of definition (access) points.

Thus, while the results obtained by the neural network approach from [39] are 1.7% worse than those of the baseline (when averaging over all 25 ESTPs), our evolutionary approach delivers an average improvement of 0.96% over the baseline which is quite considerable given the fact that the domain specific heuristic [38] only achieves an average improvement of 0.28%. In fact, DECMO2 is able to find the best solution for 22 out of the 25 ESTPs and for 13 out of the 15 ESTPs with an unknown optimum. Even though, in practice, marginal improvements can be easily negated by modeling errors and / or slight changes in assumptions, the ESTP benchmark results are encouraging as they indicate that our quite generic solving strategy can discover very good minimum Steiner tree approximations across various scenarios.

The results of the six academic CAA-Structure design scenarios are presented in Fig. 6. The solution for A1 seems rather simple but one should notice how the placement of the three Steiner points is optimized such that these points are all situated in the immediate vicinity of obstacle corners. For the test cases A2 through A6, we must first highlight that the 14 access points can be grouped in 4 clusters:

- the four points under the central obstacle structure can be reunited in a *bottom cluster*;
- the six densely packed points in the central-right part of the 2D design canvas form a *right cluster*;
- the two points above the central obstacle structure form a *top cluster*;
- the two points in the center of the 2D canvas form a *central cluster* that is surrounded on 3 sides by obstacles.

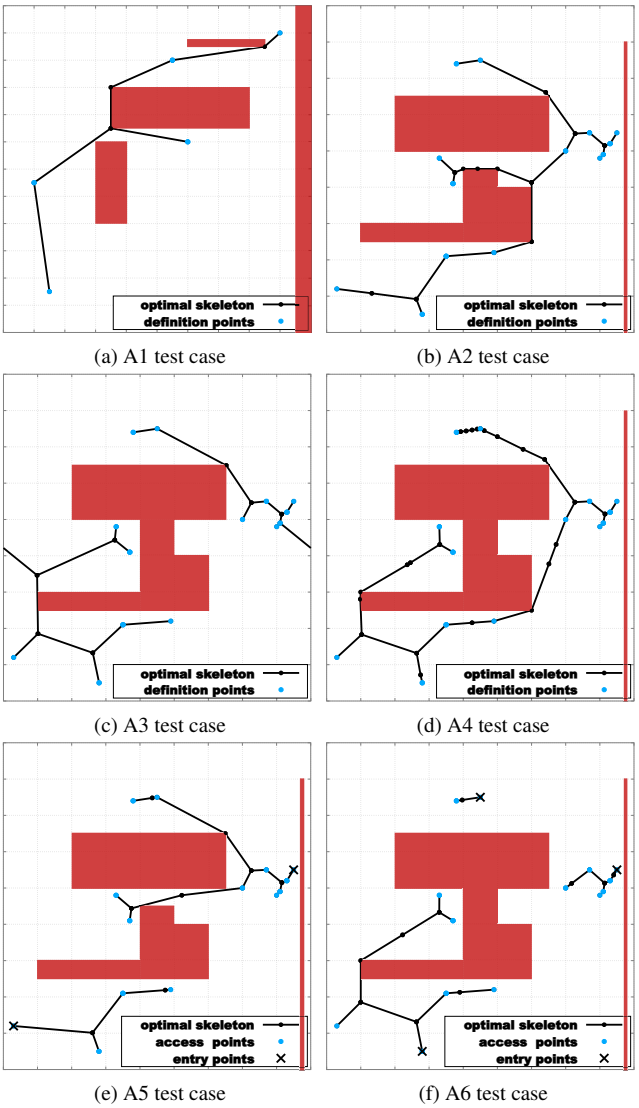
These clusters offer little room for intra-cluster path optimization and the case-specific placement of obstacle areas (and entry points in the case of A5 and A6) aims to discover if the solver is able to speculate various cost-optimal opportunities for inter-cluster connection. The results indicate that the DECMO2-based solving strategy is up to the task as:

- the solution for A2 uses a gap in the massive central obstacle structure to connect the central cluster to the path that connects the bottom cluster to the right cluster;
- the solution for A3 is able to profit from the left edge - right edge continuity (that is facilitated by the removal of the thin right-most obstacle) and still connects the central cluster to the path that connects the bottom cluster to the right cluster;
- the obstacle placement in A4 finally forces a linear link of the clusters (central $\rightarrow$ bottom $\rightarrow$ right $\rightarrow$ top) where cost gains can be made by placing Steiner points close to obstacle corners;
- the existence of two entry points in A5 (one for the bottom and one for the right cluster) facilitates a solution with a disjoint layout that does not require the longest intra-cluster paths (i.e., those connecting to the bottom cluster in A2, A3, and A4);
- the solution for A6 is able to profit from the existence of three entry points spread across the bottom, top, and

Table 1. Comparative performance of DECMO2 on ESTPs. Best results are highlighted and * marks problems with an unknown optimum.

Problem Id. [37]	n	Minimum Euclidean Steiner Tree			
		Baseline [37]	Ref. 1 [38]	Ref. 2 [39]	DECMO2
1	5	**1.6644**	**1.6644**	1.6650	**1.6644**
2A	7	**2.0776**	**2.0776**	2.0778	**2.0776**
2B	8	**2.1387**	**2.1387**	2.1393	**2.1387**
2C	6	**2.0440**	**2.0440**	2.0460	2.0441
2D	12	**2.1842**	2.2223	2.2979	**2.1842**
2G*	7	1.6018	1.5878	1.7019	**1.5594**
3	6	**1.5988**	1.6472	1.6553	**1.5988**
6*	9	1.2862	**1.2733**	1.3024	**1.2733**
11*	64	3.8380	3.8513	3.9707	**3.8274**
12*	14	1.7222	1.7222	1.7989	**1.7067**
14*	5	**1.8181**	**1.8181**	1.8300	**1.8181**
15A	5	**0.5130**	**0.5130**	0.5236	**0.5130**
16C	4	**1.1781**	**1.1781**	1.1802	**1.1781**
18*	12	1.0421	1.0332	1.0782	**1.0241**
19B*	19	2.8408	2.8567	2.9689	**2.8286**
20*	18	2.2295	2.2295	2.3248	**2.2258**
21*	19	2.1393	**2.1381**	2.1842	2.1393
24	14	**1.4248**	1.4350	1.4379	**1.4248**
24A	15	**1.4312**	**1.4312**	1.4328	**1.4312**
25*	10	1.4180	1.4180	1.4877	**1.4179**
26*	20	2.2770	**1.9767**	1.9785	1.9785
28*	16	2.3446	2.3671	2.4048	**2.3309**
29*	17	2.1974	2.1974	2.2076	**2.1869**
30*	19	1.9358	1.9358	1.9852	**1.9309**
31*	16	1.3999	1.4220	1.4343	**1.3660**

right clusters.



(a) A1 test case (b) A2 test case

(c) A3 test case (d) A4 test case

(e) A5 test case (f) A6 test case

Fig. 6. Results for the four academic test cases

5.2 Results for the LWN industrial test case

In the Liebherr mobile harbour crane scenario, when considering the four different cost settings, the best solutions discovered for the three test case variants that feature a single (ground) entry point are plotted in Figs. 7, 8 and 9.

A visual inspection of the results for each test case reveals that imposing angle-wise restrictions on the overall design of the ascent assembly can be successfully accommodated by the DECMO2-based optimization strategy. Furthermore, while angle-wise restrictions do influence the optimization outcome, the generic (star-shaped) structure that characterizes the expert CAA-Structure design is confirmed by all the cost-optimal results obtained for this somewhat simplistic test case.

As specific observations related to the discovered optimal CAA-Structure designs, it is noteworthy that:

- The less restrictive setting $z = 1$ in C3 can results in designs that apart from platforms and ladder segments also feature ramps as shown in Fig. 7c.
- The cost setting C4 is the only one that delivers solutions (e.g., Fig. 8d) that, like the expert-designed CAA-Structure illustrated in Fig. 5b, link to the top access point exclusively via staircases. This indicates that accounting for ease-of-access concerns for each definition point should be enforced in future extensions of the model abstraction.
- Shifting the ground entry point along the horizontal axis induces a negligible local effect under very restrictive angle-wise settings (Figs. 7c, 8c, and 9c) but the effect on the overall design of assembly can be larger and global when considering more degrees of freedom, as shown by the cost-optimal designs in Figs. 7a and 9a or those in Figs. 8b and 9b.

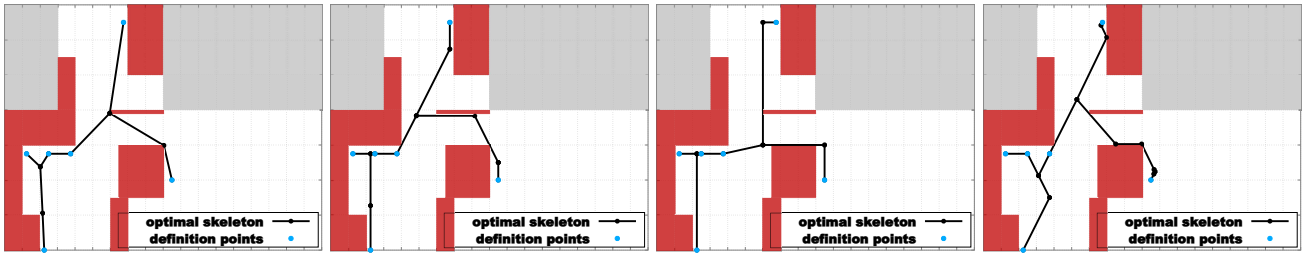
The four cost-optimal designs obtained for TC4, the scenario that considers two (ground) entry points, are shown in Fig. 10. Compared to the single entry point variants, these designs are far simpler as they are based on a disjoint layout that (in part) resembles and validates the overall form of the expert-based solution from Fig. 5b. If the decision to have a disjoint CAA-Structure would not be cost-optimal, the results from Fig. 10 would look nearly identical to those from Fig. 9 and the extra (bottom right) entry point would remain un-linked to any of the other access points.

The LWN optimization results (for MOOPs with a single primary goal) presented so far indicate that using a continuous problem formulation and a limited set of domain-dependent restrictions can yield several innovative optimal design suggestions that can in turn:

- inspire an expert / DM to consider solutions that might otherwise be overlooked;
- validate the optimality of a general design decision made by an expert – e.g., use multiple (ground) entry points in the hopes of achieving a more cost-optimal disjoint CAA-Structure.

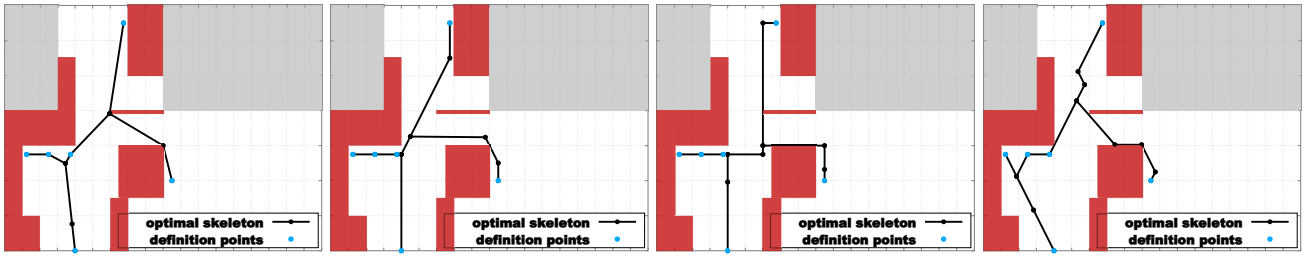
In Fig. 11 we display a filtered 2D Pareto front (notation: PF) projection of the PN discovered by DECMO2 when simultaneously optimizing LWN TC1 for the Euclidean (i.e., C1) and domain specific (i.e., C3) cost settings. As a first remark, it is noteworthy that the PF contains both extreme objective-wise optimal solutions (that best minimize C1 and C3) as well as a large set of well-spaced trade-off solutions. This is a strong indication that DECMO2 is able to discover a high-quality PN and thus successfully tackle this multi-objective optimization task that features two primary objectives.

In Fig. 11, we also illustrate four Pareto-optimal CAA-Structure designs: the two designs at the extremes of the PF and two trade-off designs. We would argue that such a Pareto representation is extremely useful to the DM as it can help



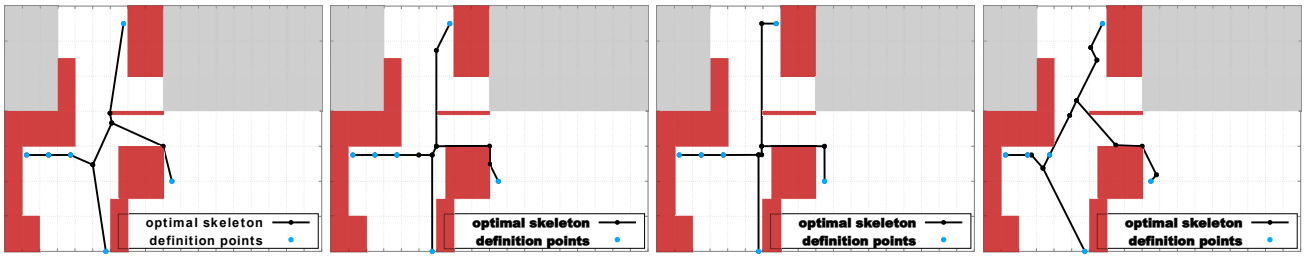
(a) C1 cost setting (b) C2 cost setting (c) C3 cost setting (d) C4 cost setting

Fig. 7. CAA-Structure optimization results for the LWN TC1 optimization scenario using different cost functions



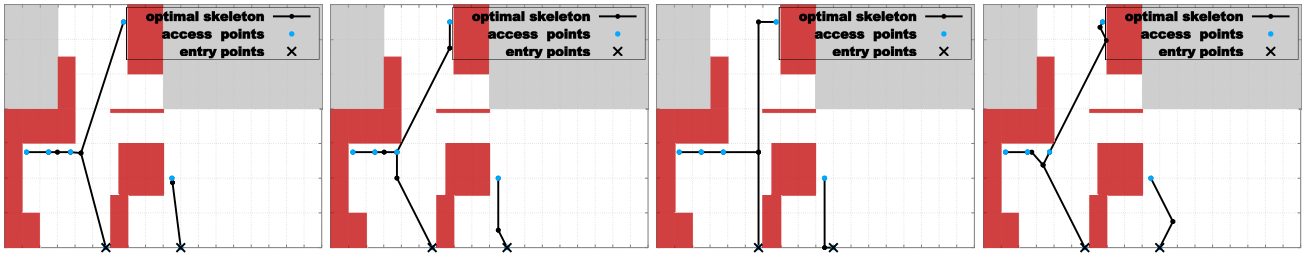
(a) C1 cost setting (b) C2 cost setting (c) C3 cost setting (d) C4 cost setting

Fig. 8. CAA-Structure optimization results for the LWN TC2 optimization scenario using different cost functions



(a) C1 cost setting (b) C2 cost setting (c) C3 cost setting (d) C4 cost setting

Fig. 9. CAA-Structure optimization results for the LWN TC3 optimization scenario using different cost functions



(a) C1 cost setting (b) C2 cost setting (c) C3 cost setting (d) C4 cost setting

Fig. 10. CAA-Structure optimization results for the LWN TC4 optimization scenario using different cost functions

to immediately identify minor "local trade-offs" (i.e., design variations) that can be worthwhile to consider. For example:

- The best solution with regard to C1 (bottom right corner) is the geometrically ideal design and has an Euclidean cost of about 1450 and a domain-specific cost of about 2425.
- The best solution with regard to C3 (top left corner) has an Euclidean cost of about 1650 and also a domain-specific cost of about 1650 (as it doesn't violate any angle-wise constraints).
- By providing access to the right-most definition point via a steep staircase / slanted ladder module, trade-off solution no.1 has an Euclidean cost of 1550 and a domain specific cost of about 1750. Thus, when comparing with the geometrical and domain-specific ideals, this solution reduces Euclidean costs by $\approx 50\%$ of the total possible since (i.e., $\frac{100}{1650-1450}$) by accepting a loss of $\approx 13\%$ (i.e., $\frac{100}{2425-1650}$) with regard to domain-specific costs.

Although trade-off solution no. 2 might prove interesting when aiming to further move towards geometric optimality, one should note that this design features 5 modules that break the domain specific restrictions (imposed by C3) of us-

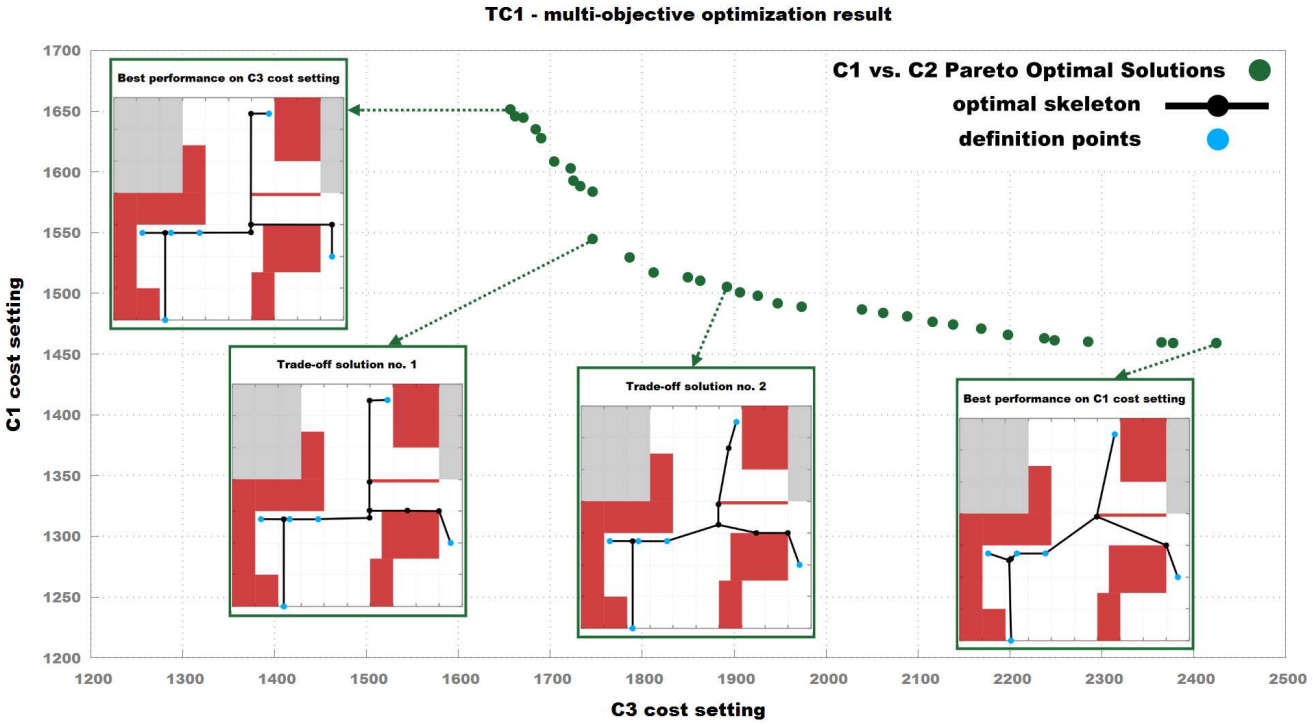


Fig. 11. Pareto optimal solutions for LWN TC1 when considering the C1 and C3 cost settings

ing only platform and vertical ladder ascent assembly modules. As such, trade-off solution no. 1 can be seen as the best compromise solution as it delivers a consistent improvement with regard to geometric optimality at the expense of a somewhat minor violation of domain specific restrictions.

6 General conclusions and outlook

In the present work we have introduced an initial, practical model abstraction for the task of automating the *cost-optimal design of complete ascent assembly structures*. In order to tackle the *2D Path network layout problem* that emerges from the aforementioned model abstraction in a domain-realistic manner, we propose a design synthesis procedure based on a multi-objective problem formulation and an advanced coevolution-based solver – DECMO2. As results obtained on benchmark and academic test cases were very encouraging, we also applied our approach on a real-life CAA-Structure design scenario provided by an industrial partner.

The results for the real-life CAA-Structure optimization scenario also empirically support the validity of our approach for both fully connected and disjoint designs. Thus, by employing appropriate cost functions, formalizing the CAA-Structure optimization problem on a continuous design space facilitates the synthesis of a wide range of innovative designs that provide engineers with valuable insight regarding the best achievable solutions and the trade-offs between the best geometrical CAA-Structure design (that requires infinite degrees of freedom and possibly new types of assembly modules) and the best practical (i.e., domain specific / currently constructible) CAA-Structure design that only requires traditionally used ascent assembly modules.

In the future we plan to investigate the hybridization potential between our current approach and a complementary design strategy [41] that is based on a discretization of the design surface and that delivers competitive results on CAA-Structure optimization scenarios with strong angle-wise restrictions.

Since the evolutionary-based search logic of our DECMO2-based solving strategy is very loosely bound to the 2D model abstraction, future work will also revolve around solving the cost-optimal design problems directly in 3D space. The reason is that the simple 2D representation is rather restrictive for several real world applications. Since the extension to 3D design scenarios could be coupled with a switch to a simulation-based fitness evaluation function (that is likely far more computationally-intensive), an analysis of the best basic parallelization options [42,43] for the resulting optimization / design synthesis scenarios might also prove very useful.

Acknowledgements

This work was supported by the K-Project “Advanced Engineering Design Automation” (AEDA) that is financed under the COMET funding scheme of the Austrian Research Promotion Agency.

References

[1] Frank, G., Entner, D., Prante, T., Khachatouri, V., and Schwarz, M., 2014. "Towards a generic framework of engineering design automation for creating complex CAD models". *International Journal on Advances in Systems and Measurements,* **7**(1), pp. 179–192.

[2] Antonsson, E. K., and Cagan, J., 2005. *Formal engineering design synthesis*. Cambridge University Press.

[3] Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N. V., and Wood, K. L., 2011. "Computer-based design synthesis research: an overview". *Journal of Computing and Information Science in Engineering,* **11**(2), p. 021003.

[4] Stöckli, F., and Shea, K., 2017. "Automated synthesis of passive dynamic brachiating robots using a simulation-driven graph grammar method". *Journal of Mechanical Design,* **139**(9), p. 092301.

[5] Zăvoianu, A.-C., Saminger-Platz, S., Entner, D., Prante, T., Hellwig, M., Schwarz, M., and Fink, K., 2017. "On the optimization of 2D path network layouts in engineering designs via evolutionary computation techniques". In EUROGEN 2017 - Book of Extended Abstracts, Technical University of Madrid.

[6] Kageura, M., and Shimada, K., 2004. "Finding the shortest path on a polyhedral surface and its application to quality assurance of electric components". *Journal of Mechanical Design,* **126**(6), pp. 1017–1026.

[7] Long, J., Zhou, H., and Memik, S. O., 2008. "EBOARST: An efficient edge-based obstacle-avoiding rectilinear steiner tree construction algorithm". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* **27**(12), pp. 2169–2182.

[8] Verhagen, P., Polla, S., and Frommer, I., 2011. "Finding Byzantine junctions with Steiner trees". In Computational approaches to movement in archaeology, pp. 73–97.

[9] Klincewicz, J. G., 1998. "Hub location in backbone/tributary network design: A review". *Location Science,* **6**(1), pp. 307–335.

[10] Smith, J. M., and Gross, M., 1982. "Steiner minimal trees and urban service networks". *Socio-Economic Planning Sciences,* **16**(1), pp. 21–38.

[11] Burdakov, O., Doherty, P., and Kvarnström, J., 2014. "Local search for hop-constrained directed Steiner tree problem with application to UAV-based multi-target surveillance". In *Examining Robustness and Vulnerability of Networked Systems*. IOS Press, pp. 26–50.

[12] Lin, C.-W., Chen, S.-Y., Li, C.-F., Chang, Y.-W., and Yang, C.-L., 2008. "Obstacle-avoiding rectilinear steiner tree construction based on spanning graphs". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* **27**(4), pp. 643–653.

[13] Hu, Y., Feng, Z., Jing, T., Hong, X., Yang, Y., Yu, G., Hu, X., and Yan, G., 2004. "FORst: A 3-step heuristic for obstacle-avoiding rectilinear Steiner minimal tree construction". *Journal of Information and Computational Science,* **1**(3), pp. 107–116.

[14] Zachariasen, M., and Winter, P., 1999. "Obstacle-avoiding Euclidean Steiner trees in the plane: an exact algorithm". *Lecture Notes in Computer Science,* **1619**, pp. 282–295.

[15] Warme, D. M., Winter, P., and Zachariasen, M., 2000. "Exact algorithms for plane steiner tree problems: A computational study". In *Advances in Steiner trees*. Springer, pp. 81–116.

[16] Gilbert, E., and Pollak, H., 1968. "Steiner minimal trees". *SIAM Journal on Applied Mathematics,* **16**(1), pp. 1–29.

[17] Garey, M. R., Graham, R. L., and Johnson, D. S., 1977. "The complexity of computing Steiner minimal trees". *SIAM journal on applied mathematics,* **32**(4), pp. 835–859.

[18] Miettinen, K., 1999. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers.

[19] Coello Coello, C., Lamont, G., and Van Veldhuisen, D., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series. Springer.

[20] Deb, K., and Jain, S., 2003. "Multi-speed gearbox design using multi-objective evolutionary algorithms". *Journal of Mechanical Design,* **125**(3), pp. 609–619.

[21] Coelho, R. F., 2013. "Co-evolutionary optimization for multi-objective design under uncertainty". *Journal of mechanical design,* **135**(2), p. 021006.

[22] Zăvoianu, A.-C., Bramerdorfer, G., Lughofer, E., Silber, S., Amrhein, W., and Klement, E. P., 2013. "A hybrid soft computing approach for optimizing design parameters of electrical drives". In *Advances in Intelligent Systems and Computing*, V. Snášel, A. Abraham, and E. S. Corchado, eds., Vol. 188 of *Advances in Intelligent Systems and Computing*. Springer Berlin Heidelberg, pp. 347–358.

[23] Kwong, W. Y., Zhang, P. Y., Romero, D., Moran, J., Morgenroth, M., and Amon, C., 2014. "Multi-objective wind farm layout optimization considering energy generation and noise propagation with NSGA-II". *Journal of Mechanical Design,* **136**(9), p. 091010.

[24] Langdon, W. B., and Poli, R., 1998. "Fitness causes bloat". In *Soft Computing in Engineering Design and Manufacturing*, P. Chawdhry et al., eds. Springer, pp. 13–22.

[25] Prim, R. C., 1957. "Shortest connection networks and some generalizations". *Bell Labs Technical Journal,* **36**(6), pp. 1389–1401.

[26] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation,* **6**(2), pp. 182–197.

[27] Zăvoianu, A.-C., Lughofer, E., Bramerdorfer, G., Amrhein, W., and Klement, E. P., 2014. "DECMO2: a robust hybrid and adaptive multi-objective evolutionary algorithm". *Soft Computing,* **19**(12), pp. 3551–3569.

[28] Zăvoianu, A.-C., Lughofer, E., Amrhein, W., and Klement, E. P., 2013. "Efficient multi-objective optimization using 2-population cooperative coevolution". In

Computer Aided Systems Theory - EUROCAST 2013, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 251–258.

[29] Zitzler, E., Laumanns, M., and Thiele, L., 2002. "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization". In Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), International Center for Numerical Methods in Engineering (CIMNE), pp. 95–100.

[30] Deb, K., and Agrawal, R. B., 1995. "Simulated binary crossover for continuous search space". *Complex Systems,* **9**, pp. 115–148.

[31] Deb, K., 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.

[32] Kukkonen, S., and Lampinen, J., 2005. "GDE3: The third evolution step of generalized differential evolution". In IEEE Congress on Evolutionary Computation (CEC 2005), IEEE Press, pp. 443–450.

[33] Robič, T., and Filipič, B., 2005. "DEMO: Differential evolution for multiobjective optimization". In International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), Springer, Springer Berlin / Heidelberg, pp. 520–533.

[34] Storn, R., and Price, K. V., 1997. "Differential evolution - a simple and effcient heuristic for global optimization over continuous spaces". *Journal of Global Optimization,* **11**(4), December, pp. 341–359.

[35] Jaszkiewicz, A., 2002. "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment". *IEEE Transactions on Evolutionary Computation,* **6**(4), pp. 402–412.

[36] Zhang, Q., Liu, W., and Li, H., 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. Tech. rep., School of CS & EE, University of Essex, February.

[37] Soukup, J., and Chow, W., 1973. "Set of test problems for the minimum length connection networks". *ACM SIGMAP Bulletin,* **15**, pp. 48–51.

[38] Beasley, J. E., 1992. "A heuristic for euclidean and rectilinear Steiner problems". *European Journal of Operational Research,* **58**(2), pp. 284–292.

[39] Bhaumik, B., 1994. "A neural network for the Steiner minimal tree problem". *Biological Cybernetics,* **70**(5), pp. 485–494.

[40] Liebherr-Werk Nenzing GmbH. `http://www.liebherr.com/en-GB/35267.wfw`. Accessed: 2017-03-06.

[41] Hellwig, M., Entner, D., Prante, T., Zăvoianu, A.-C., Schwarz, M., and Fink, K., 2017. "Optimization of ascent assembly design based on a combinatorial problem representation". In EUROGEN 2017 - Book of Extended Abstracts, Technical University of Madrid.

[42] Zăvoianu, A.-C., Lughofer, E., Koppelstätter, W., Weidenholzer, G., Amrhein, W., and Klement, E. P., 2015. "Performance comparison of generational and steady-state asynchronous multi-objective evolutionary algorithms for computationally-intensive problems". *Knowledge-Based Systems,* **87**, pp. 47 – 60. Computational Intelligence Applications for Data Science.

[43] Harada, T., and Takadama, K., 2017. "Performance comparison of parallel asynchronous multi-objective evolutionary algorithm with different asynchrony". In Evolutionary Computation (CEC), 2017 IEEE Congress on, IEEE, pp. 1215–1222.