IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 12, 2021, accepted May 21, 2021, date of publication June 3, 2021, date of current version June 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3086070

A Hybrid Multi-Population Approach to the Project Portfolio Selection and Scheduling Problem for Future Force Design

KYLE ROBERT HARRISON[1], SABER ELSAYED[1], (Member, IEEE), IVAN L. GARANOVICH[2], TERENCE WEIR[2], MICHAEL GALISTER[2], SHARON BOSWELL[2], RICHARD TAYLOR[2], AND RUHUL SARKER[1], (Member, IEEE)

[1]School of Engineering and Information Technology, University of New South Wales Canberra at the Australian Defence Force Academy, Canberra, ACT 2600, Australia

[2]Joint and Operations Analysis Division, Defence Science and Technology Group, Department of Defence, Canberra, ACT 2610, Australia

Corresponding author: Kyle Robert Harrison (kyle.harrison@unsw.edu.au)

This work was supported by the Australian Department of Defence, Defence Science and Technology under Project RG191353.

ABSTRACT Future Force Design (FFD) is a strategic planning activity that decides the programming of defence capability options. This is a complex problem faced by the Australian Department of Defence (DoD) and requires the simultaneous selection and scheduling of projects. Specifically, this is a NP-hard problem known as the Project Portfolio Selection and Scheduling Problem (PPSSP). While the PPSSP is a complex problem itself, its complexity is further increased when coupled with the additional characteristics that arise in the context of defence-oriented planning, such as long planning periods and complex operational constraints. As a result, many previous studies examined only a small number of projects over a short planning period and are largely unsuitable for the scale required in the defence sector. To address this issue, two primary contributions are made in this paper. Firstly, this study describes a complex practical PPSSP, inspired by the FFD process, and develops a corresponding mathematical model. Problem instances are derived from real-world, publicly-available defence data. Secondly, to address instances of the problem, two existing meta-heuristics are considered and a hybrid, multi-population approach is proposed. Results are compared against those attained by a commercial exact solver and indicate that there is no statistically significant difference in performance between the proposed multi-population approach and the exact solver. A key benefit of the proposed meta-heuristic approach is that its run time is not significantly influenced by the complexity of the problem instance. Additionally, many interesting practical insights regarding the solution of selection and scheduling problems are uncovered.

INDEX TERMS Future force design, capability based planning, project portfolio selection and scheduling.

I. INTRODUCTION

The Future Force Design (FFD) problem is a complex planning task undertaken by defence organizations that assists in making critical investment decisions for the future defence force. This problem is often addressed using a Capability-Based Planning (CBP) framework. CBP provides an analytical framework for delivering *capabilities*, suitable for a wide variety of challenges, within the confines of an economic framework [1]. A capability, in this context, refers to the ability to achieve an operational effect and may include various integral components such as doctrine, training, and leadership [2]. The CBP approach encourages the adoption of strategies that address a wide variety of plausible future scenarios by focusing on the development of capabilities rather than specific countermeasures [2]. Moreover, a robust set of low-level capabilities are selected such that they can be composed in different ways to meet any complex requirements that may arise in the foreseeable future. In contrast, earlier planning approaches often focused on threat-based planning or point-scenario planning, whereby single scenarios were used to guide the planning process towards the mitigation of specific plausible events [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Kai Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

The first step in a CBP framework is to define a set of high-level objectives. In the context of FFD, the 2020 Defence Strategic Update (DSU) [4], published by the Australian Department of Defence (DoD),[1] lists three primary strategic defence objectives: 1) shape Australia's strategic environment, 2) deter actions against Australia's interests, and 3) respond with credible military force, when required. These high-level objectives can then be further decomposed into sub-objectives, such as prioritization of the immediate geographical region for Australian Defence Force (ADF) deployment, enhancing the capacity to support civil authorities in response to natural disasters and crises, and developing new capabilities to provide deterrence from nuclear threats [4].

In the 2020 Force Structure Plan (FSP) [5], the DoD provided a comprehensive review of their capability plans to realign them with the new objectives given in the DSU. To support this plan, approximately \$575B in government funding has been allocated to the DoD over the next decade, with approximately \$270B of capability investments. This funding, referred to as the Defence budget, is comprised of three major categories: the acquisition of new capabilities, sustaining existing capabilities, and workforce costs. This study focuses on the first category, namely acquisition, which encompasses the process of selecting and scheduling the delivery of capabilities such as military equipment, facilities and infrastructure, and information and communications technology projects.

The FSP also outlined a proposed budget for the 10-year period starting with the 2020-21 fiscal year (FY). Specifically, the budget allocation for the acquisition category in the 2020-21 FY was approximately \$14.4B, which increases to \$29.2B in the 2029-30 FY. This represents an increase from 34% of the total Defence budget to 40%, thereby evidencing that the acquisition and delivery of new capabilities are increasingly important to the DoD. To provide further granularity, the FSP also revealed the proportion of capability investments that are allocated to the primary capability streams over the next decade. This included five major capability streams, with allocations of \$75B for Maritime, \$65B for Air, \$55B for Land, \$15B for Information and Cyber, and \$7B for Space over the next decade. These budgets, expressed as relative percentages, are shown in Figure 1.

At its core, the primary objective of the FFD process is to maximize the delivery of new capabilities through the selection and scheduling of a set of projects subject to various operational constraints. The general problem of selecting and scheduling a portfolio of projects is referred to as the project portfolio selection and scheduling problem (PPSSP). It is well-known that both the 0–1 selection problem (i.e., the knapsack problem) and the resource-constrained project scheduling problem are NP-hard. Thus, it follows that the PPSSP, which integrates both selection and scheduling, is also NP-hard. It has also been shown that considering the selection and scheduling components independently will lead

[1] Available from: https://www.defence.gov.au/strategicupdate-2020/

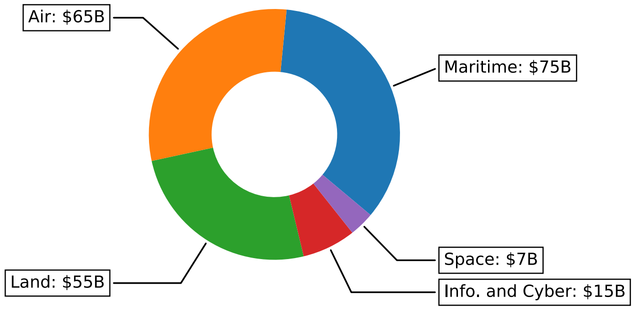


FIGURE 1. Approximate budget allocation for the five major capability streams for the 2020-30 FYs, as outlined in the 2020 FSP.

to sub-optimal solutions [6]. However, much of the work in this domain focuses on either project selection or scheduling, with a tendency to consider the former [7]. Furthermore, despite the complexity associated with solving the PPSSP, many traditional approaches to this problem make use of exact solvers, especially in the context of defence problems [8].

In addition to the typical complexities of the PPSSP, project selection and scheduling in a defence context has its own unique characteristics [9]. Specifically, defence projects typically span across a relatively long time period and have an associated yearly cost, which are characteristics not often addressed in the literature. Many studies consider only a short-term planning window, whereas defence planning must consider both the short-term and long-term implications of the implemented plan, especially considering that defence planning should be considered as an administrative tool with political repercussions, rather than a strict optimization process [10]. Furthermore, various complex operational constraints, such as a strict overall budget, often with additional yearly constraints, sub-budgets for various capability streams, referred to as "colours of money" [11], mutually exclusivity among the set of project options, and complex precedence relationships, are common in this domain. Additionally, Brown *et al.* [11] argued that "... a superficially simple annual budget constraint over a long planning horizon is ridiculous in the real world." in the context of defence-oriented planning, thereby indicating the complex nuances associated with this particular domain. Despite the unique nature of project selection and scheduling in a defence context, little attention has been given to this domain in the literature [8], [12]–[17].

It was recently identified that many applications of portfolio selection and/or scheduling in this domain made use of exact solvers [8]. While exact solvers provide definitive optimal solutions, they quickly become prohibitively expensive in terms of computation time when the problem size is increased, especially with the additional complexities present in FFD. The use of exact solvers is largely a result of a disconnect between the literature and real-world problems, whereby the problems addressed in the literature are small-scale. As a few recent examples, all published within the previous two years (i.e., 2019 and 2020), Song *et al.* [18] considered 10 projects; Dixit and Tiwari [19] considered 20 projects; Kumar *et al.* [20] considered up to 20 projects; and Song [21] considered only 6 projects. In contrast, the Canadian Navy Level 1 business planner is reported to have over 1200 projects [22]. Evidently, there is a major discrepancy between the scale of problems considered in the literature and the large-scale, real-world problems found in the context of defence planning. This paper aims to address this gap in the literature.

This paper investigates the use of meta-heuristic approaches to address a PPSSP formulated in the context of FFD, thereby bridging the aforementioned discrepancy between the literature and the real-world problems faced by defence organizations. A mathematical model, inspired by the PPSSP in the context of FFD, is proposed. The primary motivation for the model formulation is to represent a problem that closely aligns with the requirements of the FFD planning process that is conducted by the Australian DoD. Although this problem can be recognized as a PPSSP, there are additional domain specific constraints and conditions that make the problem challenging. In this paper, a novel formulation for this special PPSSP was developed with specific conditions.

A set of 20 large-scale problem instances are generated using statistical distributions that closely align with, and are directly derived from, real-world, publicly-available defence data. A hybrid, multi-population meta-heuristic approach is proposed to address the problem formulation. This hybrid approach is designed to leverage the strengths of two existing meta-heuristic approaches recently employed for a similar problem [23], namely their accuracy and speed, respectively. The proposed hybrid approach is then compared to both of the constituent meta-heuristic techniques. Furthermore, to ascertain their absolute performance relative to an exact solver, the meta-heuristic approaches are compared against solutions obtained using the Gurobi™ commercial solver.

There are two main contributions in this paper. Firstly, the introduction of a complex practical problem and the development of its mathematical model. This formulation is inspired by the requirements of the real-world FFD planning task undertaken by the Australian DoD. Secondly, the design of a specialized solution approach for solving the developed model.

The remainder of this paper is structured as follows. Section II provides background information on various applications of the PPSSP and the optimization approaches examined in this study. The mathematical model describing the proposed PPSSP formulation and the data generation process is outlined in Section III. Section IV describes the experimental procedures, the results of which are presented in Section V. Finally, concluding remarks and avenues of future work are given in Section VI.

II. BACKGROUND

This section provides a brief summary of the relevant literature on the PPSSP and an introduction to the optimization techniques employed in this study.

A. APPLICATIONS OF THE PROJECT PORTFOLIO SELECTION AND SCHEDULING PROBLEM

Ghasemzadeh *et al.* [24] proposed a 0-1 model for integrated project selection and scheduling, including various operational constraints, that was addressed with a commercial exact solver. Their example application consisted of 20 projects and an eight-year planning period. One notable limitation was that all projects must be completed within the planning horizon, thereby limiting the ability to consider defence applications, whereby the projects typically continue for a much longer period than the initial planning window.

Sun and Ma [25] proposed a “packing multiple boxes” approach to project selection and scheduling in the context of research & development projects. The study proposed an iterative approach that considered each planning period using an exact solver, which was then used to update the model for subsequent planning periods. The example application considered eight candidate projects over a five-year planning period. The iterative approach of this study may lead to sub-optimal solutions.

Liu and Wang [26] proposed a constraint programming approach for the integrated problem of project selection and scheduling with time-dependent resources and project inter-dependencies. Their empirical investigation consisted of 15 projects over a planning period of two years (i.e., 720 days), where the maximum duration for a project was 185 days.

Garcia [27] considered the PPSSP with time windows and limited inventory capacity. This study demonstrated that it is difficult to solve such problems with exact approaches and proposed a stochastic, priority-based meta-heuristic approach. The maximum number of projects considered in this study was 100.

Fisher *et al.* [28] proposed a heuristic, inspired by dynamic programming, to select optional, low-value projects for the Royal Canadian Navy. In the experimental section, this study considered a 25-year planning period with up to five new projects arriving each year.

Kumar *et al.* [20] proposed a tabu search algorithm for the simultaneous selection and scheduling of projects. The empirical investigation addressed problems with various complexities, with the highest complexity level consisting of 10–20 projects over 7–10 time periods.

The studies of Song *et al.* [21] and Song *et al.* [18] proposed heuristic algorithms based on a stochastic multi-attribute acceptability analysis approach. The case study considered in [21] was a hospital construction plan in Hefei, China consisting of six projects with a maximum project duration of 15 years. The case study in [18] was based on public housing projects in Guangzhou, China and considered 10 projects in a short-term planning period. Notably, the cost for each project in this study was limited to only a two-year period.

Dixit and Tiwari [19] proposed a model to address the PPSSP using a conditional value-at-risk approach, which was solved using a combination of a commercial simulation engine and a meta-heuristic approach for optimization. The case study, modelling the business needs of a dairy farm, considered a set of 20 projects over a 20-year planning period.

As can be seen from the above applications, many approaches consider only a relatively limited number of projects – the maximum number identified above was 100 projects. While some of the studies included the use of meta-heuristics or other non-exact solvers, none of the studies included an examination of a large-scale, real-world, defence-oriented scenario.

B. MATHEMATICAL MODELLING

Exact solvers refer to a class of mathematically-inspired optimization approaches that can derive an optimal solution for an optimization problem [29]. Optimization problems, when addressed by exact solvers, are classified by the type(s) of the decision variables (e.g., real-valued, integer, mixed, etc.) along with the characteristics of the objective function and constraints (e.g., linear, non-linear, etc.). For example, a linear program refers to the case where a linear combination of the decision variables can be used to model both the constraints and objective function. When the decision variables are restricted to integer values, as will be considered in this study, this is referred to as an integer program. An integer program that is also linear is known as an integer linear program.

It is well known that different problem types require different solution approaches and can result in drastically different algorithmic complexities. For example, a linear program over real numbers can be solved in polynomial time whereas optimizing an integer linear program is NP-hard, which means there is no known polynomial-time algorithm to solve such problems [30]. When addressing integer programs, one of the most common approaches is branch and bound, which is a technique to recursively enumerate the search space of an optimization problem using a tree structure. Candidate solutions are partitioned into subsets, referred to as *branches*, and the best known upper and lower *bounds* for the objective fitness associated with the optimal solution are maintained. Branches that are proven incapable of leading to an improvement of the bounds are discarded, thereby causing a reduction in the number of solutions that must be enumerated. Given that solving integer linear programs is NP-hard, using exact solvers for many real-world problems is computationally infeasible. In some cases, a relaxation (i.e., simplification) of the model is used to facilitate the identification of feasible solutions in a reasonable amount of time, though these solutions are likely to be sub-optimal with respect to the non-relaxed model.

C. GENETIC ALGORITHM

The Genetic Algorithm (GA) [31] is a population-based meta-heuristic inspired by the concept of Darwinian evolution. A population of candidate solutions, referred to as chromosomes, are evolved using selection, recombination, and mutation operators. The selection mechanisms are designed to provide a selection bias towards chromosomes with better objective fitness values, thereby guiding the entire population towards promising regions of the search space.

At each generation in the optimization process, candidate solutions are evaluated according to the supplied objective function and are then used to generate a new population. If desired, a proportion of the best-fit solutions are first inserted directly into the next generation via a process known as *elitism*. The elitism mechanism prevents the best-known genetic material from being lost when generating a new population. A selection mechanism is used to generate a mating pool of chromosomes, with a bias towards better-fit chromosomes. Members of the mating pool are then recombined to form offspring chromosomes via the crossover operator. Crossover operators, in general, recombine the genetic material of two or more parent solutions with the intention of generating chromosomes that exhibit a better objective fitness, increased population diversity, or both. The generated offspring may, according to the value of the mutation rate parameter, undergo a mutation process before being inserted into the new population. In general, the mutation operator performs a small perturbation on the genetic material of a chromosome to introduce new genetic material into the population. This entire process repeats until the desired termination criteria are met. The GA provides a framework such that different selection, mutation, and crossover operators can be used to formulate an optimizer that is tailored to the problem at hand.

D. BIASED RANDOM-KEY GENETIC ALGORITHM

The Random-Key Genetic Algorithm (RKGA) [32] is a real-valued variant of the GA, often employed for combinatorial optimization, based on the concept of random keys, which are an indirect representation of a candidate solution. A solution encoded with the indirect representation is referred to as a genotype, while the decoded direct representation of a candidate solution is referred to as a phenotype. Random keys in the RKGA are random numbers, $\sim U(0, 1)$, that are used during the decoding of a genotype solution into its corresponding phenotype solution. Phenotype solutions are then evaluated and the population is sorted according to the objective fitness. A fixed proportion of the best-fit solutions are moved to an elite population, while the rest remain in the non-elite population. Elitism is implemented by directly copying the members of the elite population to the next generation. Individuals are recombined by randomly selecting two members from the entire population (i.e., without regards to elite vs. non-elite solutions) and performing a uniform crossover operation. Mutation is done via immigration, which

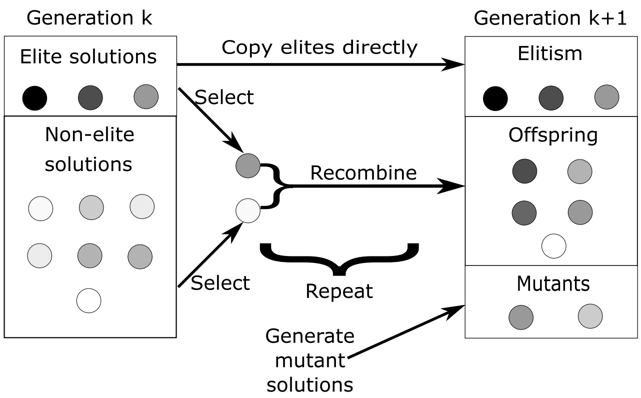


FIGURE 2. Creation of a new population in BRKGA.

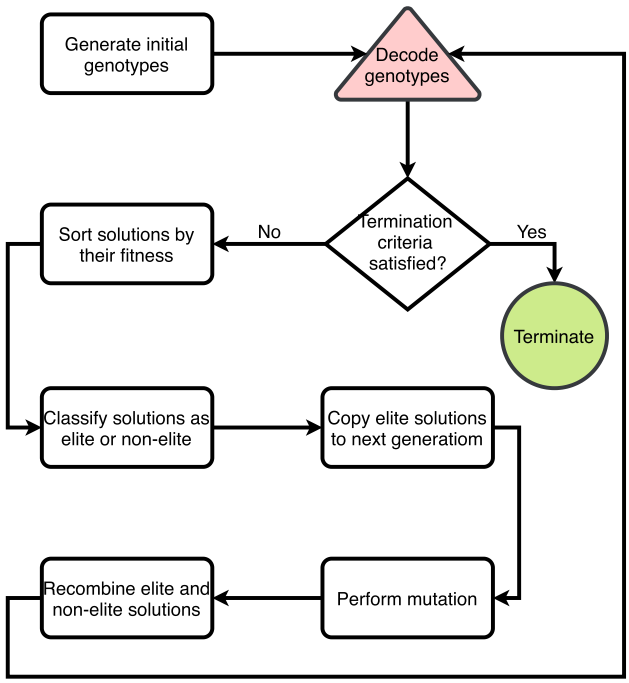


FIGURE 3. Flowchart of the evolution process in BRKGA.

refers to the generation of new random solutions using the same process as the initial population generation. With the exception of the decoding process, all steps in the RKGA are problem-independent.

The Biased Random-Key Genetic Algorithm (BRKGA) [33] differs from the RKGA by selecting one parent for crossover from the elite population and one parent from the non-elite population, then biasing the crossover in favor of the elite solution. The BRKGA has been shown to improve the performance of the RKGA, in general [33], [34]. Fig. 2 depicts the creation of a new population in BRKGA whereas Fig. 3 depicts the entire evolutionary process. Note that, the proportion of elite solutions, offspring, and mutants that make up the subsequent generation are user-supplied parameters.

E. DIFFERENTIAL EVOLUTION

Differential Evolution (DE) [35] is an evolutionary optimization algorithm that iteratively improves a population of candidate solutions, referred to as individuals. Individuals within the population, initially placed at random positions in the feasible search space, are updated using mutation, recombination, and selection operations. In the DE algorithm, trial positions, which represent potential new positions for an individual, are created through a recombination operator. Similar to a GA, different operators can be employed in DE to tailor the optimizer to the current problem being considered. The choice of operators is usually denoted by DE/s/n/c where s is the selection operator, n is the number of trial vectors to be generated, and c refers to the crossover operator.

The most common variant of DE is referred to as DE/rand/1/bin [36], whereby individuals are selected randomly, a single trial position is created, and crossover is performed using a binary operator. Creation of the trial position $\mathbf{t}$ for an individual $\mathbf{x}$ in dimension i using the DE/rand/1/bin strategy is given by

$$t_i = \begin{cases} a_i + F(b_i - c_i) & \text{if } rand() < c_r \text{ or } i = randi(D) \\ x_i & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are three randomly selected, distinct members of the population that are different from the current individual $\mathbf{x}$, $F \in [0, 2]$ is the user-supplied differential weight, $rand() \sim U(0, 1)$, $randi(D)$ selects a uniform random integer in the range $[1, D]$, D is the problem dimensionality, and $c_r \in [0, 1]$ is the user-supplied crossover probability. If the generated trial position improves the individual's fitness, the trial position is accepted and the position of the individual is updated accordingly. Otherwise, the trial position is discarded and the individual retains its current position.

III. PROBLEM FORMULATION AND DATA GENERATION

This section describes the formulation of the model used in this study as well as the process for generating synthetic data for the examined problem instances.

A. MODEL FORMULATION

The PPSSP model proposed in this paper is motivated by the CBP process carried out by the Australian DoD in the context of FFD. This model extends the formulation provided in [23] by adding capability streams, and their associated budgets, as well as the addition of budgetary constraints to limit the spending on initiating and maintaining ongoing projects, respectively. The primary objective of this process is to maximize the delivery of capabilities resulting from the selection and scheduling of projects, subject to various operational constraints. The remainder of this section uses the following notation.

- N is the number of projects considered.
- T is the number of planning periods.

- x_{it} is the (binary) decision variable and is set to 1 during the time period ($t \in T$) in which project i is scheduled to be initiated, otherwise it is set to 0.
- v_{it} is the value assigned to project i in year t of its lifetime.
- r is the time-discount factor.
- c_{it} is the cost of project i in year t of its lifetime.
- B_t is the total available budget in year t.
- CS is the set of considered capability streams.
- CB_s is the total budget available for capability stream s during the planning window.
- SB_t is the total budget available for starting projects in year t.
- OB_t is the total budget available for the maintenance of ongoing projects in year t.
- d_p is the duration of project p.
- T_B is the total budgeting period, which is defined as $T + \max(d_p)$, i.e., the number of planning periods plus the maximum project duration.
- $PC(p)$ is the set of predecessor projects that must be completed before project p can be started.
- $ME(p)$ is the set of mutually exclusive projects, of which only one from the set can be programmed.

Formally, the problem examined in this study is modelled by

(Maximize total value of projects across all years):

$$\max \sum_{i=1}^{N} \sum_{k=1}^{T} \sum_{t=1}^{k} x_{it} \frac{v_{i,k-t+1}}{(1+r)^k} \tag{2a}$$

subject to

(Budget constraints in each time period):

$$\sum_{i=1}^{N} \sum_{t=1}^{k} x_{it} c_{i,k-t+1} \leq B_k, \quad \forall k \in T_B \tag{2b}$$

(Budget constraints for each capability stream):

$$\sum_{i=1}^{N_s} \sum_{k=1}^{T_B} \sum_{t=1}^{k} x_{it} c_{i,k-t+1} \leq CB_s, \quad \forall s \in CS \tag{2c}$$

(Maximum budget for starting projects):

$$\sum_{i=1}^{N} \sum_{t=1}^{T_B} x_{it} c_{i,1} \leq SB_k, \quad \forall k \in T_B \tag{2d}$$

(Maximum budget for ongoing projects):

$$\sum_{i=1}^{N} \sum_{t=1}^{k} x_{it} c_{i,k-t+1} \leq OB_k, \quad \forall k \in 2, 3, \ldots, T_B \tag{2e}$$

(Precedence constraints):

$$\sum_{t=1}^{T} x_{jt} \geq \sum_{t=1}^{T} x_{it}, \quad \forall i \in N \text{ and } \forall j \in PC(i) \tag{2f}$$

$$\sum_{t=1}^{T} t x_{it} \geq \sum_{t=1}^{T} (t + d_j) x_{jt}, \quad \forall i \in N \text{ and } \forall j \in PC(i) \tag{2g}$$

(Mutual exclusion constraints):

$$\sum_{t=1}^{T} x_{it} + x_{jt} \leq 1, \quad \forall i \in N \text{ and } \forall j \in ME(i) \tag{2h}$$

(Each project can be scheduled a maximum of one time):

$$\sum_{t=1}^{T} x_{it} \leq 1, \quad \forall i \in N. \tag{2i}$$

Eq. (2a), the objective function, calculates the time-discounted total value of all projects in every year using a discount rate of r. Time discounting is used to reduce the value associated with a project based on the time it is scheduled to start, if applicable, as a means to prioritize earlier delivery of capabilities. Specifically, for each year in the planning period after the initial planning year, the value of a project is decreased by the discount rate r. Note that, the value of a project is used as a proxy for the delivery of capability and, therefore, Eq. (2a) represents the overall delivery of capabilities achieved by the selected portfolio and associated schedule. Eq. (2b) enforces that the cost of all current running projects must be less than the available budget in each time period under consideration. Eq. (2c) enforces a long-term cumulative budget constraint, such that the total funding allocated to each capability stream across the entire planning horizon is within its respective predefined allocation. Eqs. (2d) and (2e) ensure that upper bounds for the costs associated with starting and maintaining projects, respectively, are respected. These two constraints were added to mitigate the end effects observed in [23]. Eqs. (2f) and (2g) enforce the precedence constraints such that a project cannot be started unless all of its prerequisites have been completed. Eq. (2h) ensures that a project cannot be selected if one of the projects from its mutually exclusive group, if applicable, has been selected. Eq. (2i) ensures that a project can be selected a maximum of one time.

A simple example of the data format and a corresponding feasible schedule is shown in Fig. 4. Note that, projects 1 and 5 are mutually exclusive, indicating that only one of them may be selected for inclusion in the portfolio. In this example, project 5 has been selected, thereby preventing the selection of project 1. Additionally, project 3 is a prerequisite for project 4, indicating that project 4 must not start before project 3 has concluded. The total portfolio value (without time discounting) for this schedule is 232, as the sum of the values for projects 2, 3, 4, and 5. If a time discount factor of 1% was used, i.e., $r = 0.01$ in Eq. (2a), the total portfolio value would decrease to 229.48 due to projects 4 and 5 starting in periods 4 and 3, which correspond to values of $k = 3$ and $k = 2$, respectively. The cost profile associated with this schedule is shown in Fig. 5.

B. DATA GENERATION

The data generation process for this study was inspired by the process given in [17] and [23]. The initial available

PROJECT	DURATION	PREREQUISITE	MUTUAL EXCLUSION	VALUE	COST YEAR 1	COST YEAR 2	COST YEAR 3	COST YEAR 4	COST YEAR 5	FEASIBLE SCHEDULE 1	2	3	4	5	6
Project 1	5		Project 5	89	66	17	73	67	57						
Project 2	2			86	77	67				■	■				
Project 3	1			40	76					■					
Project 4	3	Project 3		44	55	87	21						■	■	■
Project 5	2		Project 1	62	90	29						■	■		

FIGURE 4. An example depicting the data format and a feasible schedule for a simple instance of the proposed problem.

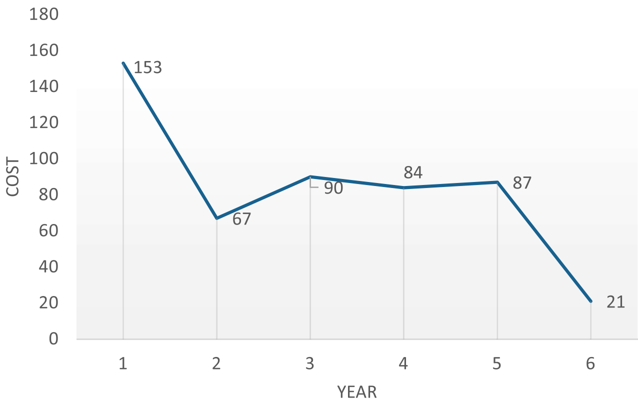


FIGURE 5. The cost profile corresponding to the schedule in Figure 4.

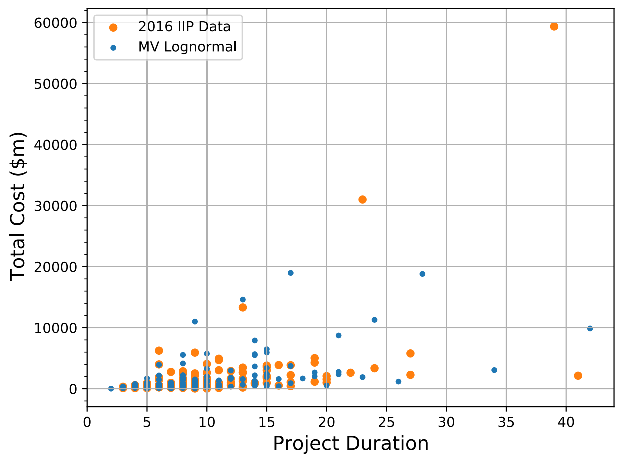


FIGURE 6. Plot of cost and duration extracted from IIP data and 148 random points generated by Eq. (3).

budget (i.e., B_1) was set at \$14,439[2] with an annual increase of \$1,637 based on information published in the 2020 FSP [5]. Five capability streams were used with the following proportions: Air (29.6%), Information and Cyber (7.4%), Land (24.7%), Maritime (34.6%), and Space (3.7%) [5]. Note that, these proportions represent the total proportion of the available budget allocated to each capability stream within the planning period, not individual yearly budget proportions. The maximum proportion of budget available for starting projects was set at 25% whereas the maximum proportion available for ongoing projects was set at 75%.

For each project, the duration (d_p) and total cost (c_p) were independently sampled from a multivariate log-normal distribution. Similar distributions have recently been shown to accurately model defence project expenditures [37]. The duration and cost for each project are sampled according to

$$\begin{bmatrix} d_p \\ c_p \end{bmatrix} \sim \text{lognormal}\left(\begin{bmatrix} 2.191054 \\ 6.642006 \end{bmatrix}, \begin{bmatrix} 0.246245 & 0.374572 \\ 0.374572 & 1.555780 \end{bmatrix}\right), \quad (3)$$

[2]Monetary figures in this study are supplied in millions.

where both c_p and d_p were rounded to the nearest integer value. The parameters for the distribution given in Eq. (3) were derived from the 2016 Integrated Investment Plan (IIP), published by the Australian DoD [38]. Fig. 6 shows the IIP data alongside 150 randomly generated points using the distribution given in Eq. (3). Furthermore, each project was randomly assigned to one of the five capability streams using the budget proportions as the respective probabilities. For example, a project was assigned to the Air capability stream with a probability of 29.6%.

Given that the distribution in Eq. (3) provides only the total cost for each project, the cost must then be distributed across the development lifetime of the project. Distributing the cost over the lifetime of a project is done by defining a cumulative distribution function (CDF) via the Weibull distribution, normalized within the range [0, 1]. The CDF then dictates the percentage of project expenditure as a function of the project completion percentage [39]. Specifically, the CDF describing the yearly expenditure of each project was generated according to

$$CDF_{c_{it}} = \frac{1 - \exp\left(\frac{t}{\beta}\right)^{\alpha}}{1 - \exp\left(\frac{1}{\beta}\right)^{\alpha}}, \quad \forall t \in 1, 2, \ldots, d_p \tag{4a}$$

where

$$\alpha \sim \mathcal{N}(1.589, 2) \tag{4b}$$
$$\beta = \max\left[\mathcal{N}(0.71, 0.3), 0.1\right]. \tag{4c}$$

It should be noted that the value of α was regenerated if it fell below 0.1. The mean values used in Eqs. (4b) and (4c) were taken from [39] and the standard deviations were determined empirically. Yearly project costs were then derived from the cumulative yearly costs provided by the CDF at each time period and were rounded to the nearest integer value. The empirical cumulative distribution plots for 10 Weibull distributions, generated according to Eq. (4), are shown in Figure 7.

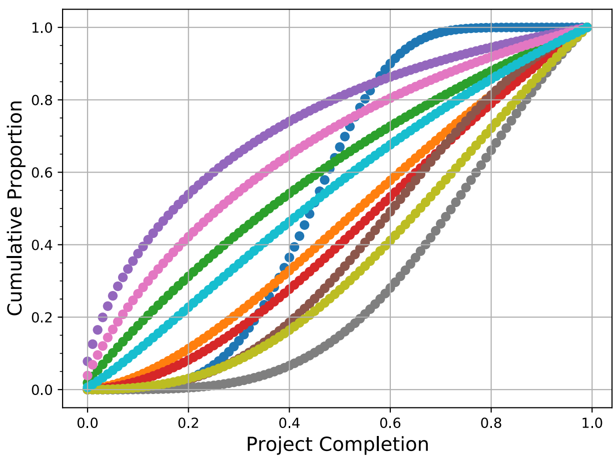


FIGURE 7. Empirical cumulative distribution plot for 10 randomly generated Weibull distributions according to Eq. (4).

Assigning the total value (i.e., v_p) for a project was done using a cost-duration valuation scheme, given by

$$v_p = U(0, 2)c_p + \sum_{j=2}^{d_p} \sim U\{1, 4\}, \tag{5}$$

where $U\{1, 4\}$ is the discrete uniform distribution with support over the range [1, 4]. The value is then distributed across the lifetime of the project using the same process as the cost, i.e., a CDF derived from the Weibull distribution given in Eq. (4). This valuation scheme reflects the empirical project data provided by Defence Science and Technology Group where the cost-value ratio generally follows a uniform distribution and the total project value depends on the project duration. This scheme is also consistent with recent literature regarding project benefit realisation [40]. Figure 8 presents a plot of the total cost and value for 150 randomly generated projects.

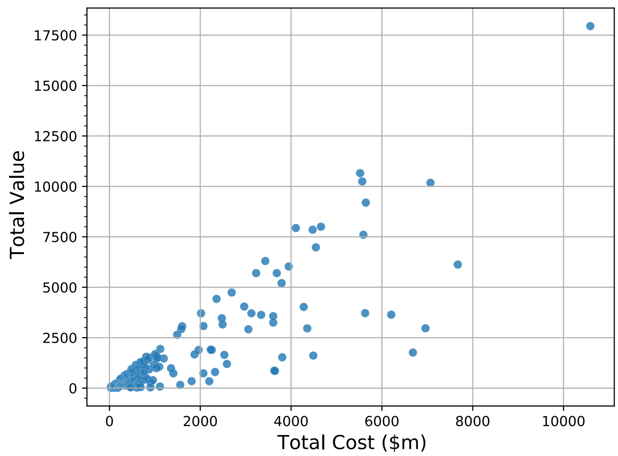


FIGURE 8. Plot of total project cost and value for 150 randomly generated projects.

To generate the set of precedences, i.e., prerequisite constraints, a fixed percentage of individuals were considered as belonging to a precedence group. Groups of size two were randomly generated, without replacement,[3] from the entire set of projects. Without loss of generality, for each generated pair of projects (i, j) with $i < j$, project i was taken as a prerequisite for project j. The set of mutual exclusion constraints were generated in an analogous fashion to the prerequisite groups, except that group sizes of both 2 and 3 were considered.

IV. EXPERIMENTAL SETUP

This section describes the experimental procedures used to demonstrate the efficacy of the meta-heuristic approaches when addressing the proposed PPSSP model.

A. META-HEURISTIC APPROACHES

Experiments were run using an 8-core, 3.6GHz Ryzen 7 1800X CPU with 32GB RAM running on Windows 10 Professional Edition.[4] Experiments were conducted using Python,[5] with many components compiled using the Cython[6] static compiler.

1) BRKGA AND DE

The two primary meta-heuristics considered were the BRKGA and DE approaches. These approaches used a real-valued genotype encoding with values in the range of [0, 1]. Genotype solutions were then decoded to their phenotype representation, i.e., permutation vectors, by sequencing the projects in (ascending) order of their corresponding value in the individual. For example, an individual with a genotype solution of [0.567, 0.329, 0.658, 0.128] would be decoded to

[3] This ensures that cyclical constraints were not constructed.
[4] Version 2004
[5] Version 3.7.7
[6] Version 0.29.21

the permutation vector [4, 2, 1, 3] such that the fourth project would be considered first for scheduling, followed by the second, first, and third projects, respectively.

The permutation vectors, i.e., phenotype solutions, represented a permutation of the values from 1 to n, where n is the number of projects, denoting the order in which projects were considered for scheduling, i.e., a priority vector. The projects were then scheduled at their earliest feasible start time. Projects that could not be feasibly scheduled were ignored. Schedule order occurs as previously described for the permutation vector. Note that, the start time of projects then depends on the available budget and problem constraints. Therefore, the permutation vector given above does not necessarily mean that, for example, project 4 will be initiated during an earlier period than project 1. An important feature of this heuristic scheduling component is that it implicitly prevents infeasible portfolios from being generated. Therefore, the meta-heuristic approaches do not require expensive constraint handling mechanisms nor do they require repair mechanisms.

Both the BRKGA and DE approaches were run until 100 iterations had occurred with no improvement to the global best fitness. Performance data is collected over 30 independent runs for each algorithm.

2) HYBRID META-HEURISTIC

A hybrid, multi-population meta-heuristic approach, that would simultaneously evolve both a BRKGA and DE population, was also employed. The primary motivation for this approach stems from a recent study that evaluated various meta-heuristic approaches on a similar PPSSP formulation [23]. The study of [23] found that, of the examined meta-heuristic approaches, BRKGA demonstrated the best performance. Preliminary experiments then determined that DE provided the fastest convergence, with only minor reductions in solution quality. Hence, the proposed approach was designed to leverage the strengths of both BRKGA and DE.

The proposed multi-population approach used a knowledge transfer mechanism to exchange information between the two sub-populations. To exchange information, the best n_e individuals from each sub-population were periodically sent to the other sub-population according to the merge frequency control parameter, f_m. The transferred individuals were merged with the existing population and the worst n_e individuals were removed to maintain consistent population sizes. As with the BRKGA and DE meta-heuristics, the hybrid approach was run until 100 iterations with no improvement to the global best fitness, taken as the best fitness attained by either population, was observed. Furthermore, results were taken over 30 independent runs.

3) GUROBI COMMERCIAL SOLVER

As a baseline for comparison of absolute performance, the Gurobi™ commercial solver (version 9.0.2, via the Python interface) was also used to solve problem instances. The Gurobi solver was run until an error gap of less than 1% was found, with a time limit of 5 minutes. Error gap, in the context of Gurobi solutions, refers to the difference between the current best feasible solution and the best maximum bound identified. Therefore, an error gap of 1% indicates that the solution returned by Gurobi is within 1% of the optimal solution. It should be noted that the error gap is dependent on both the quality of the solution and the tightness of the bound. Therefore, a non-zero error gap does not necessarily mean a sub-optimal solution was found.

To ascertain the performance of the meta-heuristic approaches relative to the exact solver, an error metric for portfolio p was calculated as:

$$e = \left(1 - \frac{f(p)}{f^*(p)}\right) * 100 \tag{6}$$

where $f(p)$ is the total portfolio value associated with portfolio p and $f^*(p)$ is the total portfolio value attained by Gurobi on the same problem instance. Error values are multiplied by 100 to report as a percentage.

B. PROBLEM INSTANCES

Firstly, a set of 10 problem instances, that considered only the yearly budget constraint, were examined. These are meant as simplified problem instances that do not exhibit many real-world characteristics. The initial budget was set at $14,439 with an annual increase of $1,637, inspired by the figures in the 2020 FSP [5]. These problem instances are referred to as the budget-constrained instances (BCI).

To ascertain the performance in a more realistic PPSSP scenario, a set of 10 heavily-constrained problem instances (HCI) were generated using the parameters given in Table 1. These problem instances contain many properties that are representative of the optimization process underlying the FFD process.

TABLE 1. Parameters used to generate the heavily-constrained problem instances. Tuples (g, p) represent that a proportion p of projects were generated in a group of size g.

Parameter	Value
Projects	1000
Planning window	20 years
Budget	$14,439 + $1,637($t$-1)
Starting budget proportion	25%
Ongoing budget proportion	75%
Prerequisites	(2, 0.10)
Mutual exclusions	(2, 0.05), (3, 0.45)
Capability streams	5
Capability stream proportions	34.6%, 29.6%, 24.7%, 7.4%, 3.7%
Time discount rate	0.01 (i.e., 1%)

C. STATISTICAL ANALYSIS

The statistical analysis procedure, as recommended by [41], consisted of Friedman's test for multiple comparisons among all methods [42], [43] with Shaffer correction [44] using a significance level of 0.05. The performance metric used in the statistical testing process is the average ranking across all corresponding problem instances, calculated using the total portfolio value averaged across 30 independent runs on each problem.

The critical difference, which denotes the difference in average rank that must be observed for the performance of two approaches to be considered significantly different, is determined and visually presented using critical difference plots. A critical difference plot is constructed by placing the considered approaches on a horizontal axis such that their positions correspond to their respective average rankings. Approaches are then grouped by a line if the difference between their average ranks was less than the calculated critical difference. Thus, approaches appearing on the left of the plot demonstrated superior performance, on average, while grouping by a line denotes that no significant difference in performance was observed among the approaches within that group. In general, the critical difference relationship is not transitive. Additionally, p-value matrices, that present the p-values from the pairwise comparisons, are also given.

V. RESULTS AND DISCUSSION

This section presents and discusses the empirical results obtained by the experimental procedures outlined in Section IV.

A. TUNING OF CONTROL PARAMETER VALUES

Control parameter values for each of the meta-heuristic approaches were tuned using the Bayesian optimization framework provided by the *scikit-optimize* package.[7] The optimization process was run for 250 function calls (i.e., parameter configurations), with the first 50 configurations selected randomly. To evaluate the performance of a parameter configuration, the fitness was taken as the average portfolio value over 5 runs, each run for a total of 50,000 function evaluations, on a heavily constrained problem instance with 1000 available projects over a 20 year planing period. Specifically, the problem instance was generated with the properties given in Table 1.

For DE, the *dither* and *jitter* parameters refer to the use of adaptive differential weight strategies. Dither refers to randomly modifying the differential weight using a uniform distribution in the range of [0, 1] and is applied either uniformly to all individuals (scalar) or independently for each individual (vector) [45], whereas jitter refers to a very small random perturbation of the differential weight applied uniformly to all individuals. Regarding the hybrid approach, the parameter configurations for the DE and BRKGA populations were set according to the optimal parameters reported in Table 2.

Additionally, the proportion of offspring for BRKGA was taken as $1 - p_e - p_m$, where p_e and p_m are the proportion of elites and mutants. Unless otherwise noted, the earliest scheduling mechanism was used. Furthermore, while non-integer values are presented with 5 significant figures, 16 significant figures were used during experimentation.

[7] Version 0.7.4

TABLE 2. Control parameter ranges used in the hyper-parameter optimization process.

Approach	Parameter	Range	Opt.
BRKGA	Pop. size	[10, 500]	327
	Bias	[0.0, 1.0]	0.36630
	Prop. elites	[0.1, 0.5]	0.21573
	Prop. mutants	[0.1, 0.4]	0.10000
DE	Pop. size	[10, 500]	83
	Diff. weight (F)	[0.0, 2.0]	0.00000
	Cross. rate (CR)	[0.0, 1.0]	0.45829
	Crossover	{bin, exp}	bin
	Selection	{rand, best}	rand
	Dither	{none, scalar, vector}	none
	Jitter	{true, false}	true
Hybrid	Frequency (f_m)	[1, 100]	100
	Num. exchange (n_e)	[1, 25]	19

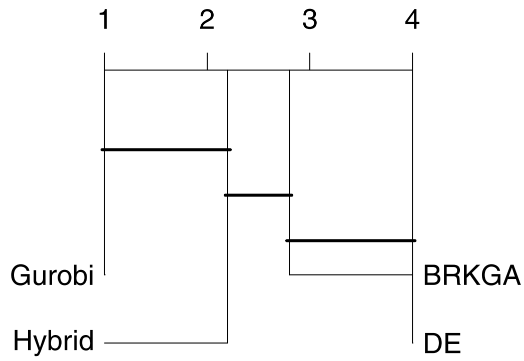


FIGURE 9. Critical difference plot for instances with only the budget constraint.

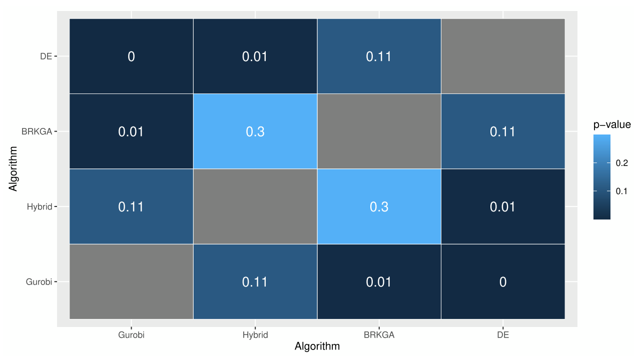


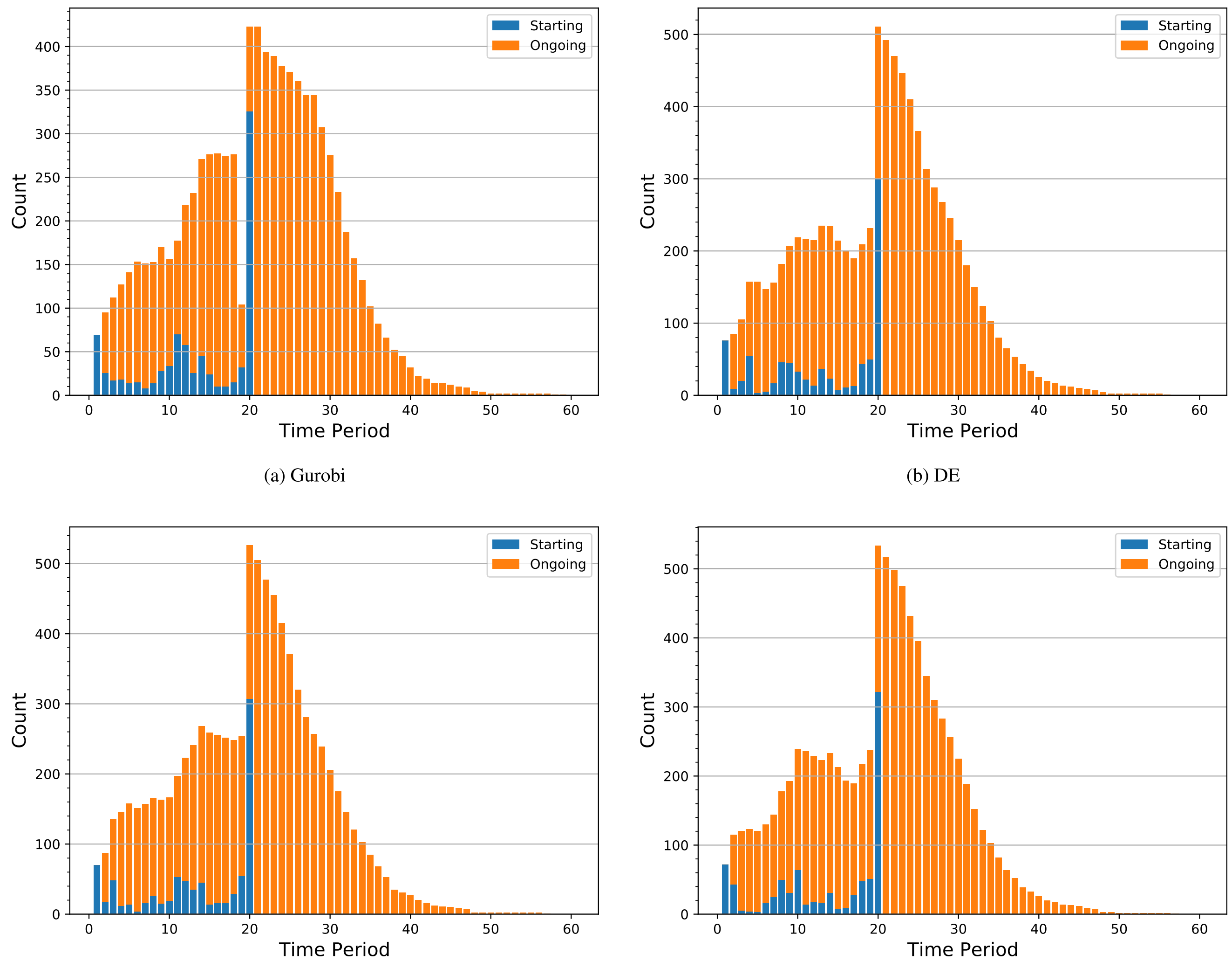
FIGURE 10. p-value matrix resulting from the Friedman test over all instances with only the budget constraint.

B. BUDGET-CONSTRAINED INSTANCES

Table 3 reports the mean, standard deviation, and minimum value of the error metric, reported as a percentage relative to the value attained by Gurobi, over 30 runs for each of the meta-heuristics on the budget-constrained instances. The results from the statistical significance tests are given via the critical difference plot in Figure 9. Furthermore, Figure 10 presents the p-value matrix, which reports the p-values associated with the pairwise comparison between the approach indicated by the row and column. Note that, the Friedman test is a two-sided test, hence the p-value matrix

TABLE 3. Mean, standard deviation, and minimum error, expressed as a percentage, on budget-constrained instances.

Instance	Gurobi	BRKGA Mean	BRKGA Std.	BRKGA Min	DE Mean	DE Std.	DE Min	Hybrid Mean	Hybrid Std.	Hybrid Min
BCI1	1565723	2.20	0.26	1.58	2.43	0.21	2.04	2.13	0.26	1.66
BCI2	1663132	3.19	0.46	2.42	3.43	0.34	2.71	3.21	0.39	2.33
BCI4	1520597	1.57	0.28	0.98	1.78	0.29	1.30	1.47	0.24	0.94
BCI5	1593587	1.94	0.22	1.58	2.17	0.26	1.70	1.84	0.19	1.52
BCI6	1681475	1.75	0.20	1.33	2.05	0.21	1.72	1.71	0.12	1.38
BCI7	1623788	1.80	0.25	1.41	2.07	0.25	1.49	1.77	0.27	1.34
BCI8	1698102	2.16	0.25	1.69	2.33	0.24	1.83	2.12	0.27	1.64
BCI9	1804336	2.64	0.32	1.95	3.03	0.28	2.54	2.52	0.24	1.92
BCI10	1660097	2.80	0.54	1.80	3.01	0.46	2.30	2.67	0.43	2.05



(a) Gurobi

(b) DE

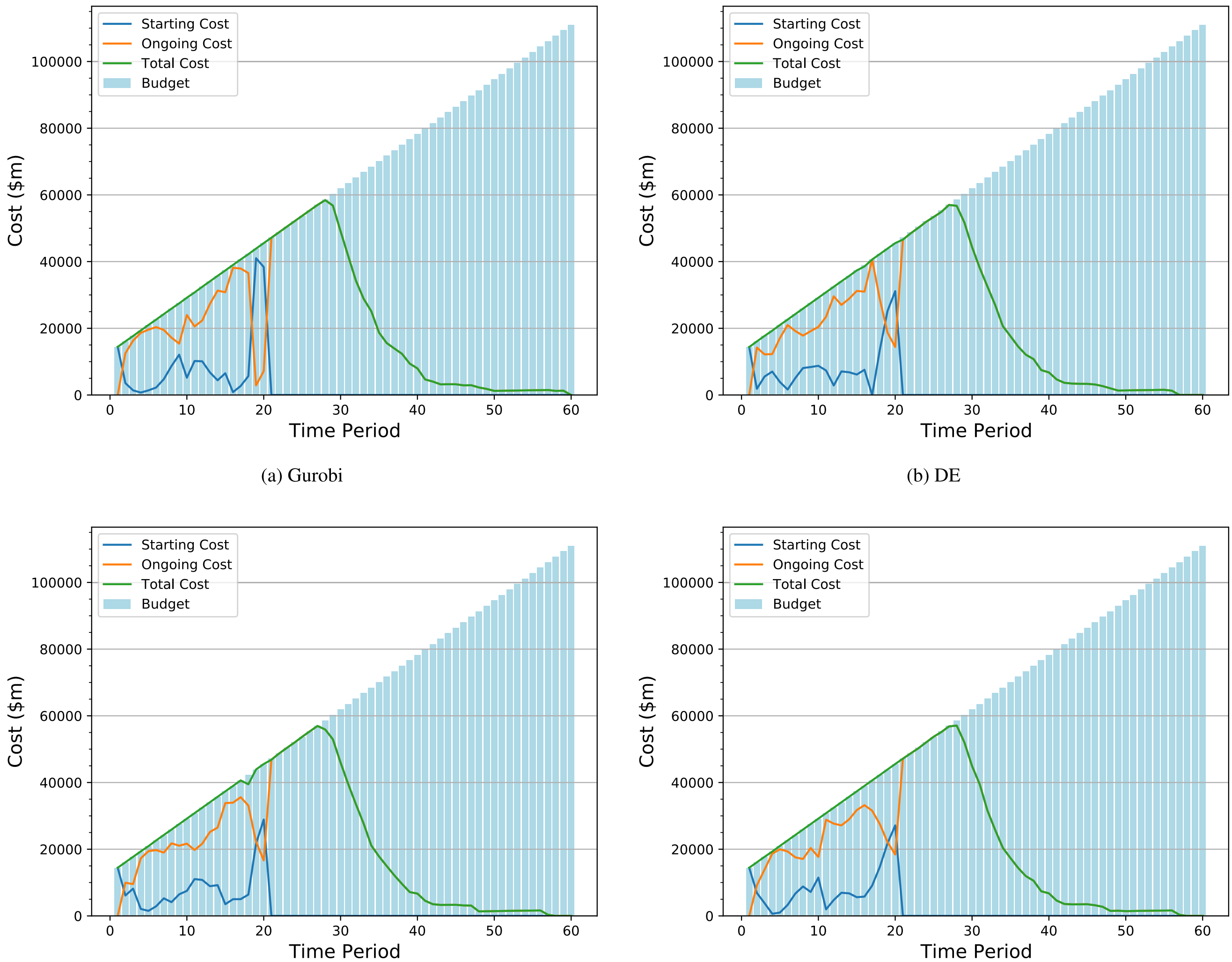
(c) BRKGA

(d) Hybrid

FIGURE 11. Number of starting and ongoing projects at each time step on the BCI1 instance.

is symmetric. It is observed from these results that both the mean error and the standard deviation was lowest for the hybrid approach, indicating the best performance of the meta-heuristic approaches. However, when considering the mean of the minimum error values, BRKGA demonstrated a slightly lower mean of 1.74% compared to the 1.79% for the hybrid approach. Additionally, the results from the hybrid approach did not exhibit a statistically significant difference when compared to the exact solver ($p = 0.11$).

The BRKGA approach depicted slightly higher average error values than the hybrid approach but the difference between the two approaches was not found to be statistically



(a) Gurobi (b) DE

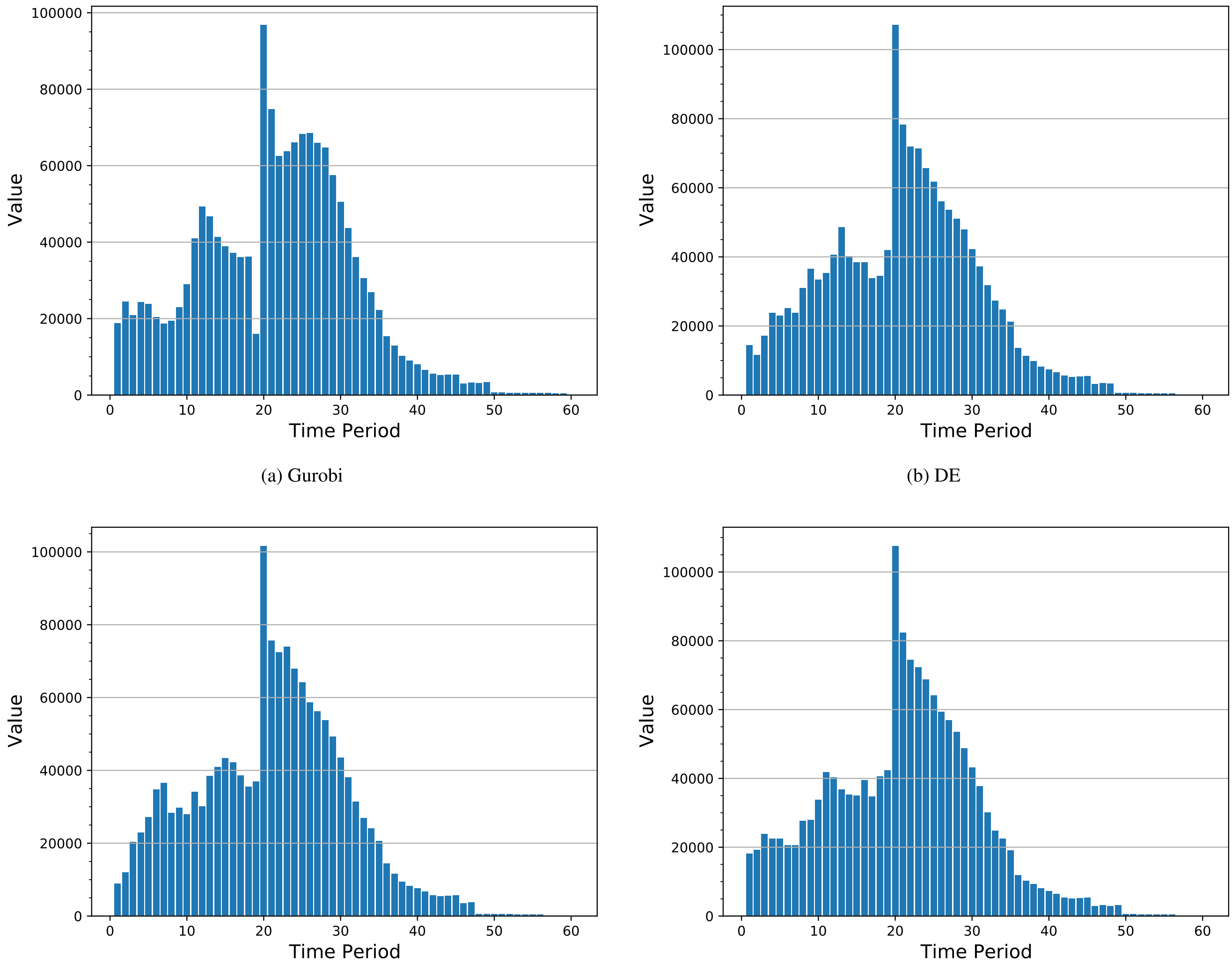
(c) BRKGA (d) Hybrid

FIGURE 12. **Costs associated with starting and maintaining projects at each time step on the BCI1 instance.**

TABLE 4. **Mean number of generations, function evaluations, and process time until termination on budget-constrained instances.**

	BRKGA			DE			Hybrid		
Instance	**Gens**	**FEs**	**Time (s)**	**Gens**	**FEs**	**Time (s)**	**Gens**	**FEs**	**Time (s)**
BCI1	945.3	243020.7	1306.8	1106.3	91825.7	457.9	1104.7	375656.7	2145.1
BCI2	974.7	250559.3	1387.0	1183.0	98189.0	497.5	1018.7	346416.7	1978.9
BCI3	1021.3	262552.7	1393.2	955.0	79265.0	376.6	983.7	334516.7	1790.6
BCI4	979.0	251673.0	1434.2	1096.7	91023.3	473.9	1012.7	344376.7	1904.5
BCI5	1064.7	273689.3	1468.2	1067.0	88561.0	448.9	1031.3	350723.3	2037.7
BCI6	1139.7	292964.3	1668.9	1096.0	90968.0	473.7	1007.7	342676.7	2002.5
BCI7	976.7	251073.3	1375.4	1088.7	90359.3	478.7	1039.0	353330.0	2107.8
BCI8	1194.7	307099.3	1623.5	1178.7	97829.3	492.8	1003.3	341203.3	1874.3
BCI9	1146.3	249677.7	1618.3	1106.7	91853.3	462.7	1019.0	346530.0	2031.9
BCI10	1064.7	273689.3	1501.2	1104.0	91632.0	479.7	997.7	339276.7	1996.5

significant ($p = 0.30$). However, due to the non-transitive nature of the critical difference, the BRKGA approach did exhibit a significant difference in performance when compared to the exact solver ($p = 0.01$). The DE approach demonstrated the worst performance overall. In the worst observed case, the average performance of DE was 5.69% worse than that of the exact solver. Overall, the difference in performance between DE and BRKGA was not significantly different when considered across all 10 problem instances ($p = 0.11$).



(a) Gurobi (b) DE

(c) BRKGA (d) Hybrid

FIGURE 13. Total portfolio value added at each time step on the BCI1 instance.

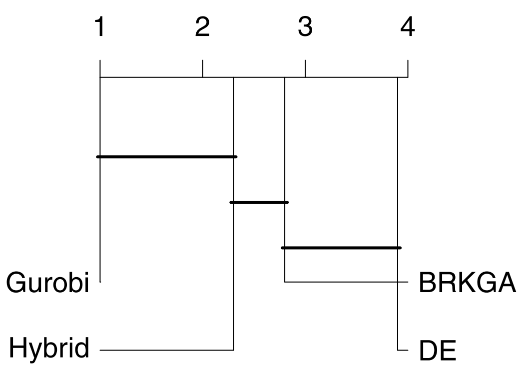


FIGURE 14. Critical difference plot for heavily-constrained instances.

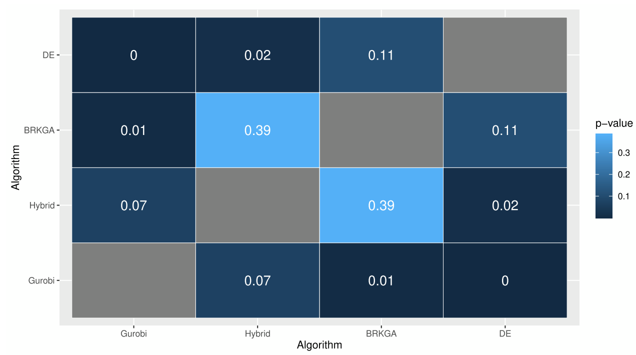
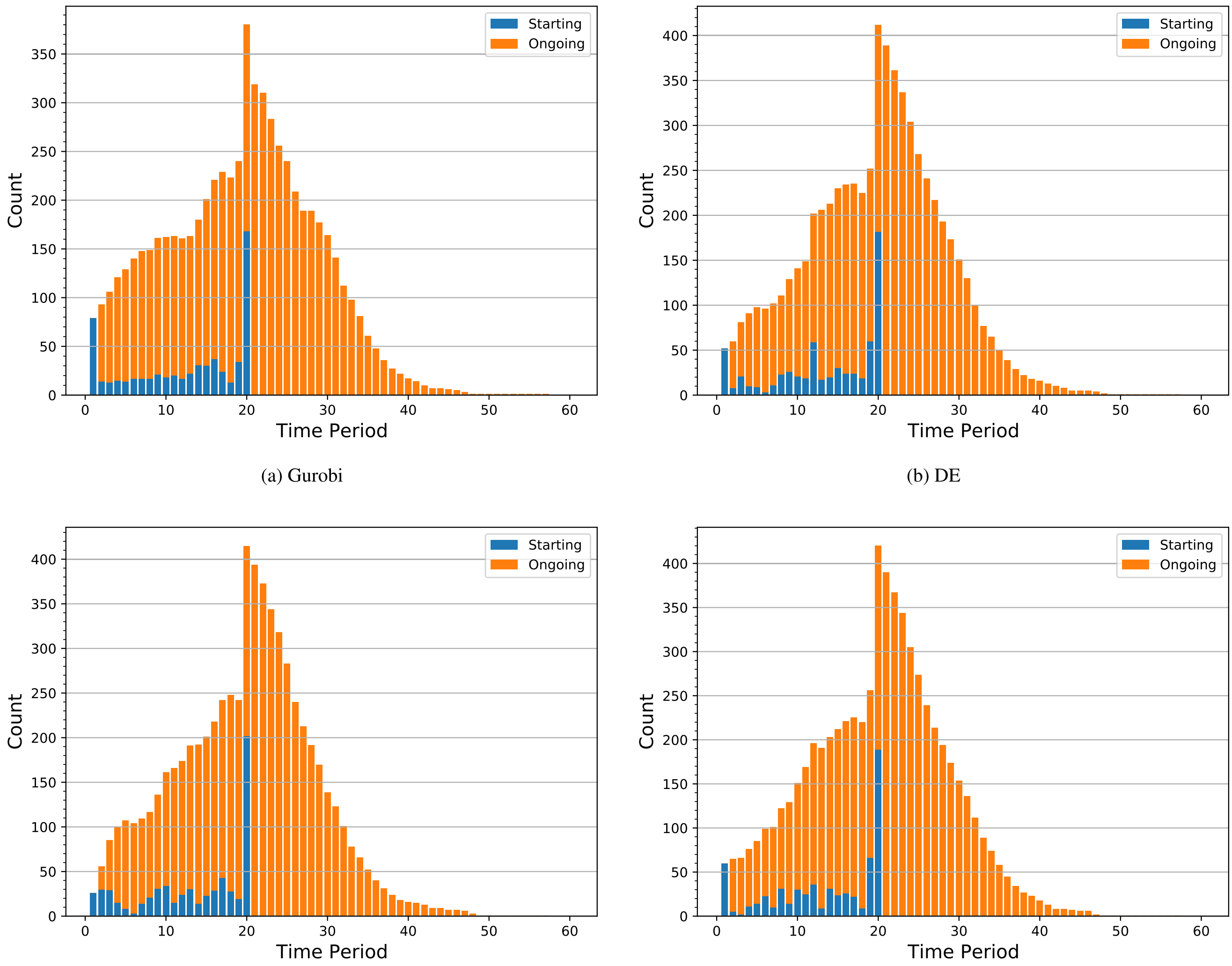


FIGURE 15. p-value matrix resulting from the Friedman test over all heavily-constrained instances.

In terms of absolute performance, compared to the exact solver, the average error for the hybrid approach ranged between 1.47% and 3.99% whereas the minimum error was observed between 0.94% and 3.18%. The worst performing approach, namely DE, exhibited a worst-case minimum error of 3.28%. Thus, it is evident that the results attained by



(a) Gurobi

(b) DE

(c) BRKGA

(d) Hybrid

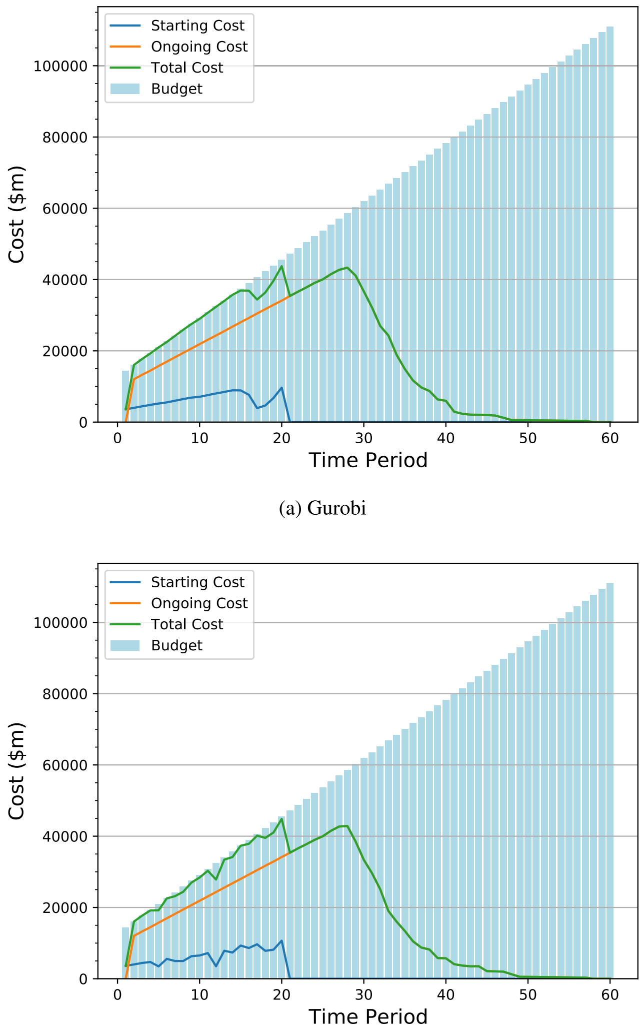
FIGURE 16. **Number of starting and ongoing projects at each time step on the HCI1 instance.**

TABLE 5. **Mean, standard deviation, and minimum value of the error metric on heavily-constrained instances.**

		BRKGA			**DE**			**Hybrid**		
Instance	**Gurobi**	Mean	Std.	Min	Mean	Std.	Min	Mean	Std.	Min
HCI1	1107896	2.80	0.37	2.06	3.09	0.36	2.41	2.89	0.54	1.99
HCI2	1164760	2.61	0.96	0.68	2.92	0.84	0.97	2.67	0.91	1.05
HCI3	1172136	3.32	0.41	2.71	3.58	0.48	2.53	3.16	0.40	2.46
HCI4	1115471	2.82	0.34	2.27	3.16	0.31	2.59	2.79	0.31	2.18
HCI5	1135912	1.65	0.37	0.82	2.12	0.31	1.51	1.57	0.34	0.94
HCI6	1204453	2.74	0.28	2.18	2.97	0.34	2.44	2.70	0.28	2.15
HCI7	1151680	2.75	0.68	1.18	2.99	0.59	1.69	2.73	0.63	1.17
HCI8	1208879	2.39	0.42	1.61	2.78	0.45	1.81	2.37	0.36	1.67
HCI9	1289799	5.15	0.49	4.37	5.69	0.47	4.51	5.29	0.43	4.55
HCI10	1165229	4.20	0.55	3.21	4.06	0.49	3.00	3.96	0.42	3.13

the meta-heuristic approaches were competitive with those attained by the exact solver.

Figure 11 shows both the number of projects starting and ongoing in each time period on the BCI1 instance. Results for the meta-heuristic approaches depict the best run for the corresponding problem instance. The first observation was that all approaches heavily back-load projects such that approximately 300 projects were started in the final time period. This is an example of an end effect [11], which occurs due to there being no special treatment of the final



(a) Gurobi

(b) DE

(c) BRKGA

(d) Hybrid

FIGURE 17. Costs associated with starting and maintaining projects at each time step on the HCI1 instance.

TABLE 6. Mean number of generations, function evaluations, and process time until termination on heavily-constrained instances.

Instance	BRKGA Gens	BRKGA FEs	BRKGA Time (s)	DE Gens	DE FEs	DE Time (s)	Hybrid Gens	Hybrid FEs	Hybrid Time (s)
HCI1	1188.7	305557.3	1779.2	1183.0	98189.0	551.6	965.3	328283.3	1926.6
HCI2	1140.7	293221.3	1697.2	1125.0	93375.0	482.7	1075.0	365570.0	2070.0
HCI3	1020.3	262295.7	1436.4	1107.0	91881.0	467.2	1075.0	365570.0	2014.1
HCI4	1093.7	281142.3	1577.2	1078.7	89529.3	474.0	1150.7	391296.7	2318.0
HCI5	1072.7	275745.3	1548.8	1087.3	90248.7	465.7	918.3	312303.3	1737.0
HCI6	1250.0	321320.0	1807.2	1077.3	89418.7	471.6	1113.0	378490.0	2162.3
HCI7	1047.0	269149.0	1509.7	1154.7	95837.3	499.1	933.7	317516.7	1810.1
HCI8	1046.3	268977.7	1482.8	1330.0	110390.0	520.3	1019.0	346530.0	2022.1
HCI9	1051.3	270262.7	1478.1	1001.7	83138.3	421.2	955.3	324883.3	1816.3
HCI10	1061.7	272918.3	1529.4	1330.3	110417.7	572.5	1044.3	355143.3	1983.8

time period. Specifically, with only the budget constraint, there was no mechanism to ensure that budget allocations after the planning window were sensible. As a result, all examined approaches leverage the fact that no new projects are initiated after the planning horizon, thereby allocating the entire budget to fund ongoing projects, as shown in Figure 12. In contrast, the meta-heuristic approaches tended to start a slightly larger number of projects in the first time period, followed by a brief lull in project initiation. This earlier start of projects led to a quicker delivery of capabilities,

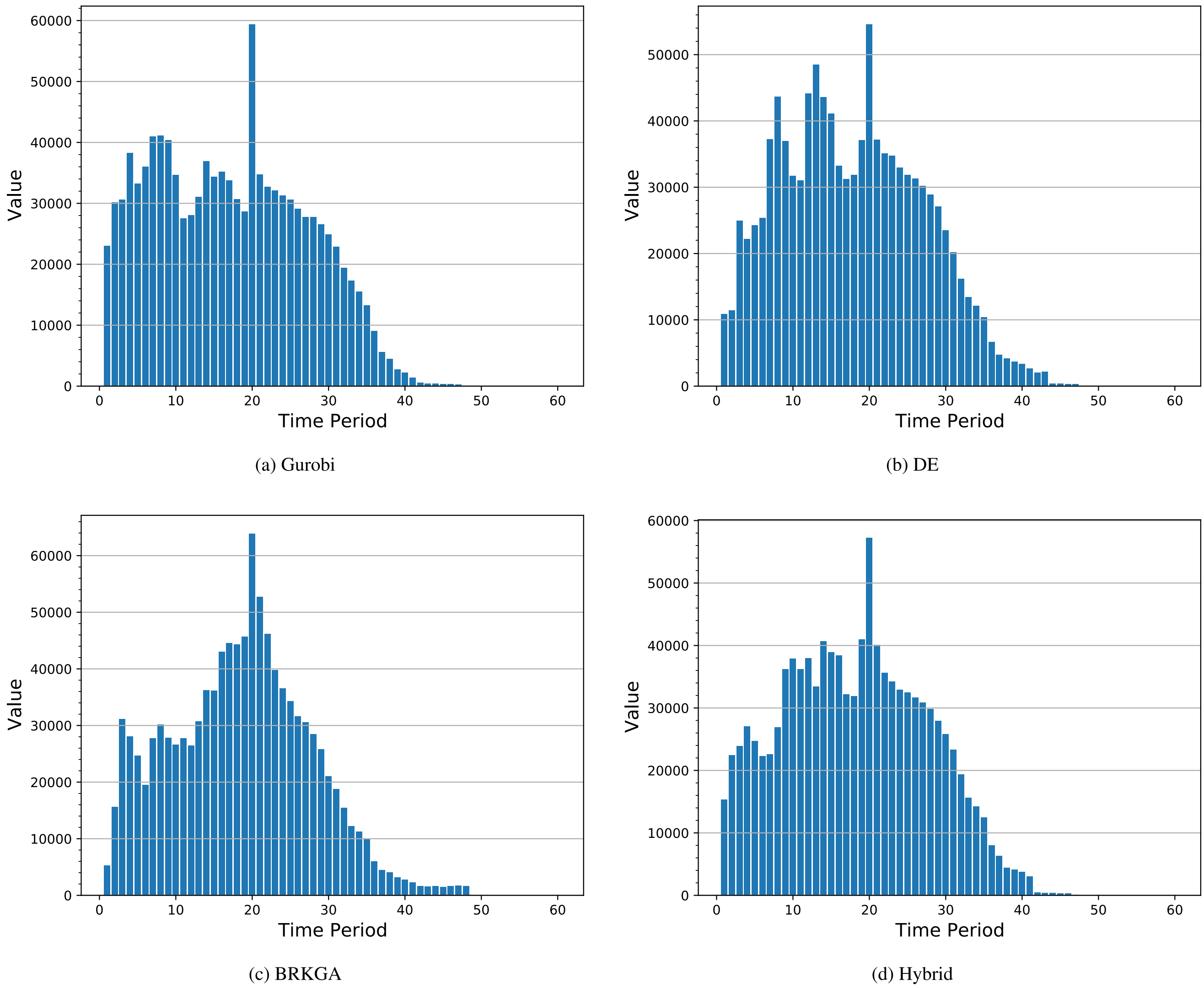
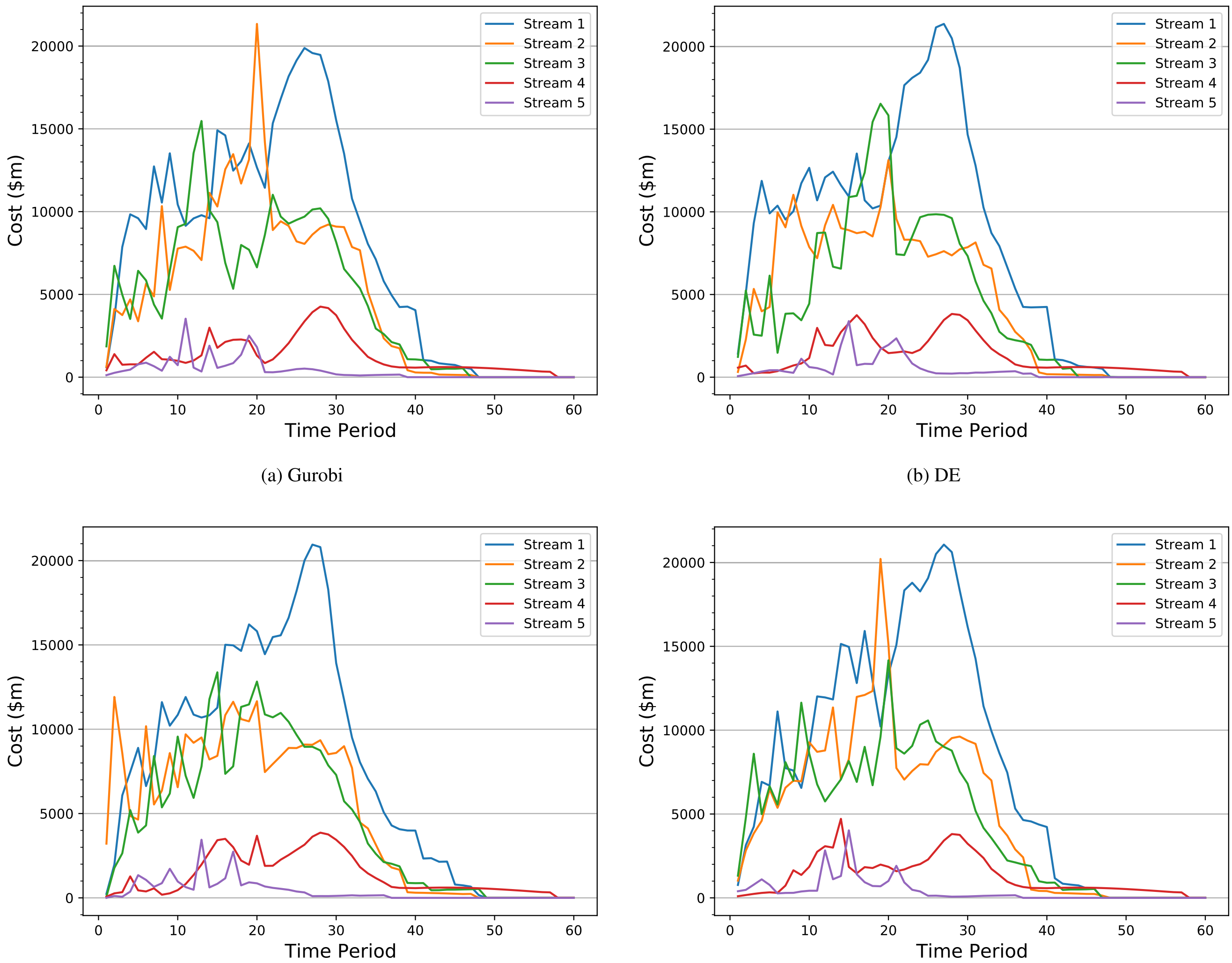


FIGURE 18. **Total portfolio value at each time step on the HCI1 instance.**

as demonstrated by the plot of value added at each time shown in Figure 13. Figure 13 also shows that the meta-heuristic approaches have a more balanced and consistent delivery of capabilities, albeit with a much larger spike at the end of the planning period, whereas the portfolio found by Gurobi exhibited multiple peaks in capability delivery, with a notable decline immediately before the end of the planning window. An additional observation was that for all examined approaches, the delivery of capabilities extended for approximately 30 years after the planning window.

To compare the running time of the meta-heuristic approaches, Table 4 gives the average number of generations (Gens), function evaluations (FEs), and process time until termination. However, a few explicit notes regarding the values presented in Table 4 are warranted. Firstly, recall that the termination criterion was set as 100 generations with no improvement to the fitness and, as a result of the different population sizes, the number of function evaluations are somewhat misleading. Secondly, the algorithms were implemented using a multi-process approach and the run time was calculated as the difference between start time and completion time for each process. Moreover, the run time is heavily dependent on the implementation, which may be inadvertently optimized more heavily for one particular approach. Therefore, the run-time is not generally indicative of the true run time for each approach. Rather, the average run time is more indicative of the overall time taken for all 30 runs, though should not necessarily be interpreted in this manner. Therefore, the measurements provided in Table 4 should be taken relative to each other, rather than as absolute measurements.

Considering the notes above, a few key observations can be made from Table 4. The number of generations required for the hybrid approach to reach convergence was slightly lower, in general, than the other two approaches. However, this result is a bit more nuanced in that the population size



(a) Gurobi

(b) DE

(c) BRKGA

(d) Hybrid

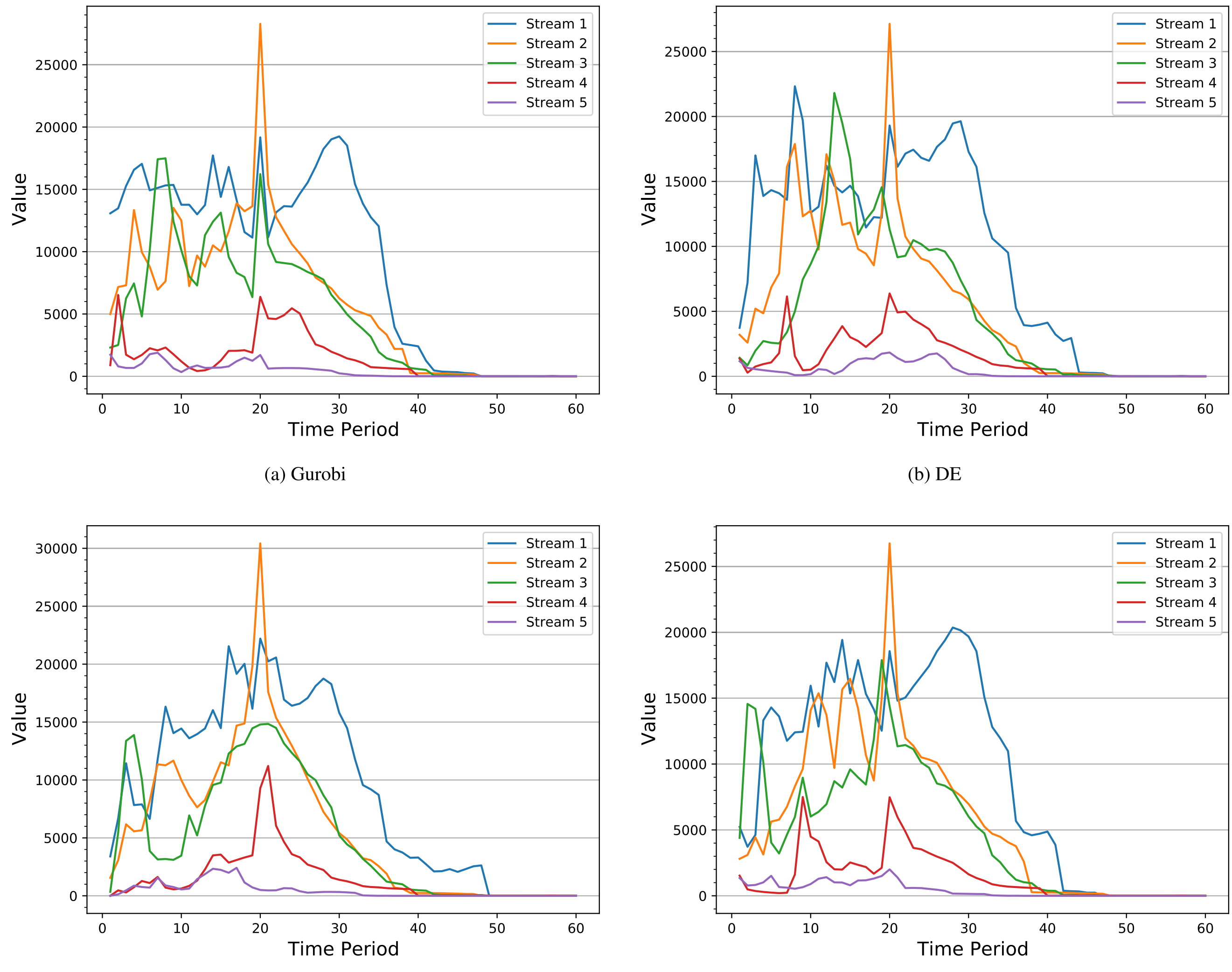
FIGURE 19. Total cost by capability stream at each time step on the HCI1 instance.

is the sum of the population sizes for both the BRKGA and DE approaches. As a result, the running time of the hybrid approach was significantly longer; the run time of the hybrid approach was approximately 1.5x longer than BRKGA and 4-5x longer than DE. This observation makes the choice of optimization approach, among those considered, much less clear than if considering performance alone. Specifically, the hybrid approach takes 4-5x longer than the DE approach, but the absolute difference in performance was generally within 0.5%. Moreover, the BRKGA approach depicted even closer performance to the hybrid approach, but took approximately 66% of the run time.

C. HEAVILY-CONSTRAINED INSTANCES

Table 5 reports the mean, standard deviation, and minimum value of the error metric over 30 runs for each of the meta-heuristic approaches on the heavily-constrained problem instances. Figures 14 and 15 present the results of the statistical significance testing via the critical difference plot and the p-value matrix, respectively. As with the lesser-constrained instances, the hybrid approach demonstrated the best performance of the meta-heuristics and was found to have no statistically significant difference in performance when compared to the exact approach ($p = 0.07$). However, the relatively low p-value of 0.07 demonstrates that the relative performance of the hybrid approach, as compared to the exact solver, was degraded as a result of the increased difficulty of the problem instances.

As observed with the simpler instances, the BRKGA exhibited slightly worse performance than the hybrid approach while the DE approach performed the worst overall. However, the difference in performance between the hybrid and BRKGA approaches was not statistically significant ($p = 0.39$). Similarly, the performance of the BRKGA and DE approaches were not found to have a statistically significant difference ($p = 0.11$). Again, the BRKGA demonstrated



(a) Gurobi

(b) DE

(c) BRKGA

(d) Hybrid

FIGURE 20. Total portfolio value by capability stream at each time step on the HCI1 instance.

a lower mean of minimum errors, at 2.11%, than the hybrid approach, at 2.13%. Nonetheless, only the hybrid approach was found to have no statistically significant difference when compared to the exact solver ($p = 0.07$).

Considering the absolute performance, compared to the exact solver, the hybrid approach exhibited a mean error of between 1.57% and 5.29%, with a minimum error between 0.94% and 4.55%. However, it should be noted that the results for all meta-heuristic approaches on the HCI9 instance were strikingly poor when compared to the exact solver. Excluding the HCI9 problem instance, the largest mean error for the hybrid approach was 3.96% while the worst-case minimum error was 3.13%. In contrast, the DE approach, which was again the worst overall, depicted a mean error between 2.12% and 5.69% (or 4.06%, excluding HCI9) whereas the minimum error was between 0.97% and 4.51% (or 3.00%, excluding HCI9).

To further examine the differences in performance, Figures 16 to 20 provide an in-depth profile of the portfolio from the exact solver and the best results from each meta-heuristic approach on the HCI1 problem instance. Figure 16 shows the number of projects that were started and ongoing in each time period. Notably, the end effect observed in the less-constrained instances (see Figure 11c) was partially mitigated by the introduction of various constraints. Specifically, limiting the proportion of the budget available to maintain ongoing projects ensured that the budget outside of the planning window was not allocated solely to the maintenance of projects. As can be seen in Figure 17c, the proportion of budget used to maintain ongoing projects continued to be capped at 75%, even outside the planning window, thereby permitting the initiation of new projects. Furthermore, the number of projects initiated in the final planning period was less than 200 – a significant decrease from the approximately 300 projects started in the final period when these constraints were not considered. Another interesting observation was that, despite the limited budget for starting projects, the number of projects initiated in the first planning period was approximately the same. This indicates that lower-cost projects, also likely shorter in duration, were prioritized in the presence of heavily-constrained environments.

Figure 18, which shows the value added at each time period on the HCI1 instance, demonstrates that the delivery of capabilities was much more consistent in the constrained instances compared to the instances with only budget constraint. Specifically, the distributions depicted a much higher delivery of capabilities in the earlier years with a far less prominent peak at the end of the planning window. One notable observation was that the portfolio returned by Gurobi demonstrated a significantly higher value in the first year of the planning period whereas the meta-heuristic approaches generally led to better delivery of capabilities in the middle of the planning period.

Finally, Figures 19 and 20 show the cost and value added at each time period by capability stream. These figures demonstrate that the delivery of capability in each stream was spread across the entire planning window. As expected, the capability stream with the highest costs and value added was stream 1, which had the largest proportion of the budget. Another noteworthy observation was that the value added by capability stream 2 in the final planning period had a large peak for all approaches.

To compare the running time of the meta-heuristic approaches, Table 6 gives the average number of generations (Gens), function evaluations (FEs), and process time until termination on the heavily constrained problem instances. The same notes as mentioned in Section V-B apply to these convergence results. In general, the observations were quite similar to those made from Table 4. Specifically, the hybrid approach required fewer generations to reach convergence, in general, than the other approaches but took approximately 1.5x as long as BRKGA and 4-5x as long as DE in terms of overall run time. The results once again indicated that the DE was able to attain relatively similar performance compared to the hybrid approach, though with a statistically significant difference, at a fraction of the running time. Another noteworthy observation was that despite the significantly increased complexity introduced as a result of the additional constraints, the run time required for convergence on the heavily constrained instances was not significantly higher than when considering the simpler instances. This can largely be attributed to the heuristic scheduling component that implicitly handles the problem constraints, thereby eliminating the need for an explicit constraint handling technique or repair mechanism. Furthermore, this implies that the time needed to reach a reasonable solution is not heavily influenced by the problem difficulty – this is an important and noteworthy property of the proposed meta-heuristic approaches.

VI. CONCLUSION AND FUTURE WORK

This study investigated the PPSSP in the context of FFD. Three meta-heuristic approaches, namely DE, BRKGA, and a hybrid, multi-population approach, were examined on two sets of problem instances. The meta-heuristics incorporated an underlying scheduling heuristic that implicitly addresses operational constraints, thereby reducing the problem complexity. To ascertain their absolute performance, the meta-heuristic approaches were also compared against a commercial exact solver. The results indicated that the proposed hybrid approach exhibited no statistical difference in performance when compared to the exact solver. While the DE approach exhibited a statistically significant difference in performance when compared to the hybrid approach, it was found to provide solutions with only a slight degradation in quality, but took 4-5x less overall run time. In absolute terms, the performance of DE was typically less than 0.5% worse than the hybrid approach and within 5% of the fitness attained by the exact solver.

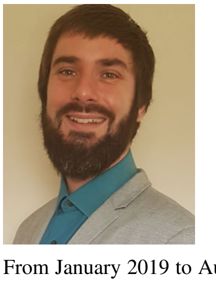
In summary, the key novelties of this study were twofold. Firstly, this study investigated a formulation of the PPSSP heavily inspired by the FFD process. Furthermore, the data generation process used statistical distributions derived from real-world defence data. Secondly, a hybrid, multi-population meta-heuristic approach was proposed to address a set of large-scale PPSSP problems that exhibited characteristics found in real-world defence project scheduling. This approach was found to exhibit no statistically significant difference in performance compared to an exact solver.

There are several avenues for expansion on this study. An effective approach to mitigating the observed end effects could be to extend the planning window beyond the desired planning period, and then only considering projects initiated within the planning period. Advantage may also be gained by allowing small budget variations and then addressing the violations by either reducing the budget allocation in future years or through a penalization mechanism. Another exploration could investigate over-programming as a hedging strategy to prevent underspends and schedule delays. Analyses could also include applying uncertainty in project costs, duration, and values as well as exploring multi-objective approaches.

REFERENCES

[1] P. Davis, "Analytic architecture for capabilities-based planning, mission-system analysis, and transformation," RAND Corp., Santa Monica, CA, USA, Tech. Rep. MR-1513-OSD, 2002.
[2] S. De Spiegeleire, "Ten trends in capability planning for defence and security," *RUSI J.*, vol. 156, no. 5, pp. 20–28, Oct. 2011.
[3] S. Hiromoto, "Fundamental capability portfolio management," Ph.D. dissertation, Pardee RAND Graduate School, Santa Monica, CA, USA, 2013.
[4] *2020 Defence Strategic Update*, Dept. Defence, New Delhi, India, 2020, pp. 1–63.
[5] *2020 Force Structure Plan*, Dept. Defence, New Delhi, India, 2020.
[6] M. A. Coffin and B. W. Taylor, "R&D project selection and scheduling with a filtered beam search approach," *IIE Trans.*, vol. 28, no. 2, pp. 167–176, Feb. 1996.

[7] A. A. Tofighian and B. Naderi, "Modeling and solving the project selection and scheduling," *Comput. Ind. Eng.*, vol. 83, pp. 30–38, May 2015.
[8] K. R. Harrison, S. Elsayed, I. Garanovich, T. Weir, M. Galister, S. Boswell, R. Taylor, and R. Sarker, "Portfolio optimization for defence applications," *IEEE Access*, vol. 8, pp. 60152–60178, 2020.
[9] D. M. Tate and P. M. Thompson, "Portfolio selection challenges in defense applications," Inst. Defense Analyses, Alexandria, VA, USA, Tech. Rep. NS D-8493, 2017.
[10] H. Breitenbauch and A. K. Jakobsson, "Defence planning as strategic fact: Introduction," *Defence Stud.*, vol. 18, no. 3, pp. 253–261, Jul. 2018.
[11] G. G. Brown, R. F. Dell, and A. M. Newman, "Optimizing military capital planning," *Interfaces*, vol. 34, no. 6, pp. 415–425, Dec. 2004.
[12] J. Xiong, K.-W. Yang, J. Liu, Q.-S. Zhao, and Y.-W. Chen, "A two-stage preference-based evolutionary multi-objective approach for capability planning problems," *Knowl.-Based Syst.*, vol. 31, pp. 128–139, Jul. 2012.
[13] J. Xiong, J. Liu, Y. Chen, and H. A. Abbass, "A knowledge-based evolutionary multiobjective approach for stochastic extended resource investment project scheduling problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 742–763, Oct. 2014.
[14] K. Shafi, S. Elsayed, R. Sarker, and M. Ryan, "Scenario-based multi-period program optimization for capability-based planning using evolutionary algorithms," *Appl. Soft Comput.*, vol. 56, pp. 717–729, Jul. 2017.
[15] J. Xiong, Z. Zhou, K. Tian, T. Liao, and J. Shi, "A multi-objective approach for weapon selection and planning problems in dynamic environments," *J. Ind. Manage. Optim.*, vol. 13, no. 3, pp. 1189–1211, 2017.
[16] M. Li, M. Li, K. Yang, B. Xia, and C. Wan, "A network-based portfolio optimization approach for military system of systems architecting," *IEEE Access*, vol. 6, pp. 53452–53472, 2018.
[17] K. R. Harrison, S. Elsayed, T. Weir, I. L. Garanovich, M. Galister, S. Boswell, R. Taylor, and R. Sarker, "Multi-period project selection and scheduling for defence capability-based planning," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 4044–4050.
[18] S. Song, T. Wei, F. Yang, and Q. Xia, "Stochastic multi-attribute acceptability analysis-based heuristic algorithms for multi-attribute project portfolio selection and scheduling problem," *J. Oper. Res. Soc.*, vol. 2020, pp. 1–17, Feb. 2020.
[19] V. Dixit and M. K. Tiwari, "Project portfolio selection and scheduling optimization based on risk measure: A conditional value at risk approach," *Ann. Oper. Res.*, vol. 285, nos. 1–2, pp. 9–33, Feb. 2020.
[20] M. Kumar, M. L. Mittal, G. Soni, and D. Joshi, "A tabu search algorithm for simultaneous selection and scheduling of projects," in *Harmony Search and Nature Inspired Optimization Algorithms* (Advances in Intelligent Systems and Computing), N. Yadav, A. Yadav, J. C. Bansal, K. Deep, and J. H. Kim, Eds., vol. 741. Singapore: Springer, 2019, pp. 1111–1121.
[21] S. Song, F. Yang, and Q. Xia, "Multi-criteria project portfolio selection and scheduling problem based on acceptability analysis," *Comput. Ind. Eng.*, vol. 135, pp. 793–799, Sep. 2019.
[22] W. Hurley, J. Brimberg, and B. Fisher, "Risk-analytic approaches to the allocation of defence operating funds," *J. Defense Model. Simul., Appl., Methodol., Technol.*, vol. 10, no. 3, pp. 275–282, Jul. 2013.
[23] K. R. Harrison, S. Elsayed, T. Weir, I. L. Garanovich, R. Taylor, and R. Sarker, "An exploration of meta-heuristic approaches for the project portfolio selection and scheduling problem in a defence context," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Dec. 2020, pp. 1395–1402.
[24] F. Ghasemzadeh, N. Archer, and P. Iyogun, "A zero-one model for project portfolio selection and scheduling," *J. Oper. Res. Soc.*, vol. 50, no. 7, pp. 745–755, Jul. 1999.
[25] H. Sun and T. Ma, "A packing-multiple-boxes model for R&D project selection and scheduling," *Technovation*, vol. 25, no. 11, pp. 1355–1361, Nov. 2005.
[26] S.-S. Liu and C.-J. Wang, "Optimizing project selection and scheduling problems with time-dependent resource constraints," *Autom. Construct.*, vol. 20, no. 8, pp. 1110–1119, Dec. 2011.
[27] C. Garcia, "A Metaheuristic algorithm for project selection and scheduling with due windows and limited inventory capacity," *Kybernetes*, vol. 43, nos. 9–10, pp. 1483–1499, Nov. 2014.
[28] B. Fisher, J. Brimberg, and W. Hurley, "An approximate dynamic programming heuristic to support non-strategic project selection for the royal Canadian navy," *J. Defense Model. Simul., Appl., Methodol., Technol.*, vol. 12, no. 2, pp. 83–90, Apr. 2015.
[29] R. A. Sarker and C. S. Newton, *Optimization Modelling: A Practical Approach*, 1st ed. Boca Raton, FL, USA: Taylor & Francis Group, 2008.
[30] K. Genova and V. Guliashki, "Linear integer programming methods and approaches—A survey," *Cybern. Inf. Technol.*, vol. 11, no. 1, pp. 3–25, 2011.
[31] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
[32] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA J. Comput.*, vol. 6, no. 2, pp. 154–160, May 1994.
[33] J. F. Gonçalves and M. G. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *J. Heuristics*, vol. 17, no. 5, pp. 487–525, Oct. 2011.
[34] J. F. Gonçalves, M. G. C. Resende, and R. F. Toso, "An experimental comparison of biased and unbiased random-key genetic algorithms," *Pesquisa Operacional*, vol. 34, no. 2, pp. 143–164, Aug. 2014.
[35] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[36] K. V. Price, "An introduction to differential evolution," in *New Ideas in Optimization*. New York, NY, USA: McGraw-Hill, 1999, pp. 79–108.
[37] T. Weir, "Regression models for project expenditures," in *Data and Decision Sciences in Action* (Lecture Notes in Management and Industrial Engineering). Cham, Switzerland: Springer, 2018, pp. 271–290.
[38] *2016 Integrated Investment Program*, Dept. Defence, New Delhi, India, 2016.
[39] T. Weir and K. Johns, "Longitudinal models for project expenditure plans," in *Proc. 22nd Int. Congr. Model. Simul.*, Dec. 2017, pp. 695–701.
[40] O. Zwikael, I. L. Garanovich, T. Weir, P. Gowlett, S. Aljoofri, and A. Ween, "Identifying triggers for new projects and programs in the Australian department of defence," in *Proc. Austral. Inst. Project Manage. Nat. Conf.*, 2018, pp. 29–38.
[41] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
[42] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
[43] M. Friedman, "A comparison of alternative tests of significance for the problem of *m* rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 1940.
[44] J. P. Shaffer, "Modified sequentially rejective multiple test procedures," *J. Amer. Stat. Assoc.*, vol. 81, no. 395, pp. 826–831, 1986.
[45] D. Dawar and S. Ludwig, "Differential evolution with dither and annealed scale factor," in *Proc. IEEE Symp. Differ. Evol. (SDE)*, Dec. 2014, pp. 1–8.



KYLE ROBERT HARRISON received the B.Sc. and M.Sc. degrees in computer science from Brock University, St. Catharines, Canada, in 2012 and 2014, respectively, and the Ph.D. degree in computer science from the University of Pretoria, Pretoria, South Africa, in 2018. He is currently a Research Associate with the School of Engineering and Information Technology, University of New South Wales Canberra at the Australian Defence Force Academy, Canberra, Australia. From January 2019 to August 2019, he was a Postdoctoral Fellow with the Department of Electrical, Computer, and Software Engineering, University of Ontario Institute of Technology, Oshawa, Canada. His research interests include computational intelligence, optimization, operations research, complex networks, and fitness landscape analysis.

SABER ELSAYED (Member, IEEE) received the Ph.D. degree in computer science from the University of New South Wales at the Australian Defence Force Academy, Canberra, Australia, in 2012. He is currently a Senior Lecturer with the School of Engineering and Information Technology, University of New South Wales at the Australian Defence Force Academy. His research interests include evolutionary algorithms, constraint-handling techniques for evolutionary algorithms, scheduling, big data, and cybersecurity using computational intelligence. He was the winner of different IEEE-CEC competitions. Since 2019, he has been serving as the Chair for the IEEE Computational Intelligence Society (ACT Chapter). He serves as an organizing committee member of different conferences in the evolutionary computation field.

IVAN L. GARANOVICH is currently a senior analyst specializing in strategic analysis of national security problems.

TERENCE WEIR is currently a senior analyst specializing in defence resource management and national security problems.

MICHAEL GALISTER has broad experience across many areas of defence operations analysis as applied to force design and experimentation and simulation in the context of both single service and joint forces.

SHARON BOSWELL is currently a senior mathematician with research interests in strategic decision analysis, particularly discrete optimization problems.

RICHARD TAYLOR is currently the Head of joint force analysis with the Australian Defence Science and Technology Group. He has extensive experience in the application of network analysis methods to mathematical modeling and operations research problems.

RUHUL SARKER (Member, IEEE) received the Ph.D. degree from Dalhousie University, Halifax, NS, Canada, in 1992. He is currently a Professor with the School of Engineering and Information Technology and the former Director of Faculty Postgraduate Research, University of New South Wales at Canberra, Canberra, Australia. He has authored the book entitled *Optimization Modelling: A Practical Approach* (Boca Raton, in 2007) and published over 300 refereed articles in international journals, edited books, and conference proceedings. His current research interests include evolutionary optimization and applied operations research. He is currently an Associate Editor of *Memetic Computing Journal*, the *Journal of Industrial and Management Optimization*, and *Flexible Service and Manufacturing Journal*.