Multiobjective Evolutionary Algorithm for Software Project Portfolio Optimization

Thomas Kremmel  
Institute of Software Technology and Interactive Systems  
Favoritenstr. 9/188  
A-1040 Vienna, Austria  
thomas.kremmel@gmail.com

Jiří Kubalík  
Department of Cybernetics  
CTU Prague  
Technická 2, 166 27 Prague 6  
Czech Republic  
kubalik@labe.felk.cvut.cz

Stefan Biffl  
Institute of Software Technology and Interactive Systems  
Favoritenstr. 9/188  
A-1040 Vienna, Austria  
stefan.biffl@tuwien.ac.at

ABSTRACT
Large software companies have to plan their project portfolio to maximize potential portfolio return and strategic alignment, while balancing various preferences, and considering limited resources. Project portfolio managers need methods and tools to find a good solution for complex project portfolios and multiobjective target criteria efficiently. However, software project portfolios are challenging to describe for optimization in a practical way that allows efficient optimization. In this paper we propose an approach to describe software project portfolios with a set of multiobjective criteria for portfolio managers using the COCOMO II model and introduce a multiobjective evolutionary approach, mPOEMS, to find the Pareto-optimal front efficiently. We evaluate the new approach with portfolios choosing from a set of 50 projects that follow the validated COCOMO II model criteria. Major results are: the proposed project portfolio management approach was found usable and useful; the mPOEMS algorithm shows capabilities for efficiently solving the combinatorial optimization problem of this type.

1. INTRODUCTION
Project Portfolio Management (PPM) is a set of processes used to support a business in conducting the mix of projects, which best fit the organisation’s various needs. This set of processes includes the selection of projects an organisation conducts, maintaining the selected projects in portfolios, and periodically reviewing the mix of projects, to check whether the selection still supports the main business goals. Various portfolio processes have already been presented. Most of them do cover the whole life cycle of portfolios, from selecting projects and optimizing portfolios to phase gate review steps [1] [15].

Portfolio optimization is a process in PPM that create the best mix of projects, out of all potential candidates. Selection of projects, and optimization of the project selection can be conducted either manually or automatically. Manual approaches to select projects are for example the Analytic Hierarchy Process(AHP), Q-Sort, scoring models, and portfolio matrices [1]. Commonly used manual approaches are based on some sort of direct comparison and ranking of the alternatives based on project data and individual preferences. Manually conducting the selection of projects is restricted in the number of projects it can deal with, constraints and objective preferences which can be taken care of, as well as the number of objectives the decision makers are able to optimize. This is especially true, since the complexity grows exponentially with the number of available project alternatives [5]. The project selection problem is NP-hard problem [5] thus there is no exact algorithm that solves larger instances of this problem to proven optimality. Hence, heuristic algorithms are an option for finding at least approximate solutions to the optimal ones.

The complexity of the project selection problem is based on the often high number of projects from which a subset has to be chosen, various existent restrictions and a multitude of objective preferences which the optimal portfolio should adhere to. Common goals of portfolio optimization are maximization of potential revenue and strategic alignment, as well as minimization of negative synergy effects in-between projects selected for a portfolio. Furthermore, portfolios often have to be balanced regarding characteristics of the portfolios. These balancing requirements are found in the number of projects assigned to a specific category, or in the optimization of a portfolio regarding its overall mean risk value.

Manual approaches to this demanding decision problem are restricted in their usefulness for the problem at hand. Promising alternatives to manually selecting projects are found in the field of mathematically based portfolio optimization, and multiobjective optimization. Automated approaches are superior to manual approaches in a way that they are able to create a set of efficient portfolios, for which it can be assured that there exist no solutions in the search space which promises better values in at least one of the objectives, and offers at least the same in all the others. This set of efficient solutions is commonly referred to as the Pareto-optimal front. Furthermore, the utilised computational power and sophisticated multiobjective meta-heuristics ensure that the solutions found do have at least the same quality as manually created portfolios. With a high probability, solutions on the Pareto-optimal front found by automated approaches are much better than solutions found with manual approaches. Furthermore, manual approaches will never be able to keep up with the search ability of automated approaches, and the number of considered projects, objectives, restrictions and constraints.

This paper presents the first phase of a decision support
framework, which focuses on the selection of projects, using an evolutionary optimization algorithm. An implementation of a nature-inspired optimization framework is presented, using the evolutionary algorithm mPOEMS, based on general goals which the portfolio management should support. This optimization framework is able to find efficient portfolios on or close to the Pareto-optimal front.

mPOEMS is an iterative optimization algorithm that seeks in each iteration for such a modification of the current solution, called prototype, that improves its quality the best. Modifications are represented as sequences of elementary actions (simple mutations in standard EAs), defined specifically for the problem at hand, mPOEMS uses an evolutionary algorithm to search for the best action sequence that can be considered a structured hypermutation.

We have chosen mPOEMS since it was recently shown to be suitable for solving discrete multiobjective optimization problems. In [9] mPOEMS was shown to produce better or at least competitive results to the state-of-the-art MOEAs such as NSGA-II and SPEA2 on the multiobjective knapsack problem. Moreover, a single-objective variant of the algorithm (POEMS) was successfully deployed for solving hard optimization problems such as the sorting network problem [10], the multiple sequence alignment problem [11], or the shortest common supersequence problem [12].

For the proof-of-concept experiments a test set of fifty IT projects was used. COCOMO II is used to calculate the effort and schedule data of the utilized project pool. The optimization framework claims to consider a very broad range of portfolio optimization characteristics and goals found in the literature. Results presented in this paper show, that mPOEMS is able to generate high quality solutions with respect to the given set of optimization criteria.

The rest of the paper is structured as follows. The next section provides a survey of project portfolio management research. In section 3, the project portfolio optimization as the multiobjective optimization problem is formally defined. Section 4 gives a brief intro into the multiobjective optimizations. Section 5 describes the proposed approach based on the mPOEMS algorithm. Section 6 describes the experimental setup and analyzes the achieved results. Section 7 concludes the paper.

2. RELATED WORK

Various approaches have already proven the applicability of mathematically based meta-heuristics for the project selection problem.

In 2000, Ghasemzadeh and Archer [6] presented a decision support system, which follows the process steps of Archer’s integrated framework for PPM [1], and combine them with a formal model, considering various variables and constraints. The possibility to making manual adjustments is given through adding or removing projects. A very interesting contribution is the consideration that projects can start in an arbitrary timeframe, given that the project is completed within the planning horizon.

In 2003, Stummer and Heidenberger presented a three-phase approach [18] for PPM in the field of research and development. With the use of an Integer Linear Programming Model, they provided a process to deal with non-numerically restricted project inter-dependencies, Logical & Strategic Model, they provided a process to deal with non-numerically completed within the planning horizon.

In 2004, Doerner and Gutjahr published a so-called Pareto Ant Colony Optimization Approach [5]. Artificial ants construct valid project portfolios and take into account complex project interactions. The consideration of project synergy, the good results it has shown in experiments and the possibility of easily adding heuristic information to the algorithm makes it a valuable approach [5].

In 2005, Medaglia, Graves and Ringuet presented an evolutionary approach for project selection [14]. They utilised a genetic algorithm, combined with random parameter values to model project selection under uncertainty. To model uncertainty is of specific relevance, because it is often difficult in the project selection stage to define parameter values exactly, due to various internal and external influences (risk, political issues, interest rates, economic changes, and so on). Uncertainty could be modelled as specific random values following triangular, exponential, and Erlang distributions like in [14], or with the use of likely values as proposed in [1].

Each of the above presented approaches add valuable information and knowledge to the topic of creating optimized project portfolios. The approaches have been analysed and inspired the authors of this paper in creating an evolutionary-based approach to the project selection problem.

3. FORMAL MODEL

The presented project selection problem is similar to the multiobjective knapsack problem as used in [19]. The designated goal of the original multiobjective knapsack problem is to find a subset of items for an arbitrary number of knapsacks, which maximize the profit for each knapsack, while the maximum capacity of all knapsacks under consideration may not be exceeded. A solution is represented by a vector with the length of the maximum available items, where position i is set to 1 if item i is selected, and 0 if not. Here, an optimal set of projects from a pool of all potential projects is sought just like the set of items in the knapsack problem. Additionally, the information assigned to each project not only specifies whether the project is selected (xi > 0) or not (xi = 0), but also specifies when the project starts (value of xi greater than 0). The planning horizon of a portfolio is divided into timeframes, and timeframes are in turn subdivided into smaller units. These units can be chosen arbitrarily. Months are used here for these units, and one year for the timeframes. The project start is declared by selecting a month for each project from the available planning horizon.

Formally, the task of the project selection problem is defined as follows:

Find a vector \( \mathbf{x} = (x_1, x_2, \ldots, x_p) \in M_1 \times \cdots \times M_p \), where \( M_i \subseteq M, M = \{0, 1, 2, \ldots, T \times 12\} \), such that \( y = (q_1(x), q_2(x), \ldots, q_p(x)) \) is maximum, where xi is greater than 0 if project i is selected, and 0 if
not, \( p \) is the number of projects in the candidate pool, \( M \) the number of months in the planning horizon, \( M \) the months in which project \( i \) can start, \( T \) the number of timeframes in the planning horizon, and \( q_i(x) \) are the five measures of portfolio quality (optimization objectives).

The list of optimization objectives is based on a thorough investigation of the corresponding literature and discussions with experts in the field of portfolio management. They are defined as follows

1. **Potential Revenue** \( (q_1(x)) \). The project portfolio is in fact an investment portfolio. An organisation invests human resources, knowledge and money into a project, with the goal of attaining benefits from this investment. Like the shares for an investment portfolio, the potential projects for the project portfolio have to be evaluated in regard to their potential financial revenue. Thus, this objective deals with the need to maximize potential overall portfolio return. It is calculated as
   \[
   q_1(x) = \sum_{i=1}^{p} f_i(x)
   \]
   , where \( f_i(x) = r_i \cdot w_i \) and \( r_i \) is the potential financial revenue of project \( i \), and \( w_i \) is 1 if \( x_i > 0 \), and 0 if \( x_i = 0 \). The greater the overall potential revenue, the better the solution.

2. **Strategic Alignment** \( (q_2(x)) \). This objective deals with the strategic alignment of projects an organisation conduct. The strategy of an organisation has been defined as a set of approaches to attain certain goals on the business level. According to Reyck [16], organisations at the lowest level of PPM adoption, face problems with commitment of business leaders, poor alignment of projects to strategy, little coordination between projects and conflicting project objectives. The selection of projects under consideration of strategic alignment of the selected projects is of major importance for a business, because the nature and direction of an organisation is defined by the conducted activities [7]. Thus, the strategic alignment on the portfolio level should be maximized. We calculate the overall strategic alignment as
   \[
   q_2(x) = \sum_{i=1}^{p} s_i(x)
   \]
   , where \( s_i(x) = a_i \cdot w_i \) and \( a_i \) is the strategic alignment value of project \( i \). The greater the overall strategic alignment value, the better the solution.

3. **Resource Usage Distribution Metric** \( (q_3(x)) \). This objective is introduced in order to create selection pressure towards the selection of portfolios which on the one hand have a high resource usage per timeframe, and on the other hand have the best distribution between the timeframes. Its value is between 0 and 1, where 1 means full resource consumption in each timeframe, and 0 means that at least in one timeframe no resources are consumed. Consequently, the objective function to maximize, may be expressed as,
   \[
   q_3(x) = \prod_{t=1}^{T} \left( \sum_{o=1}^{o} \sum_{k=1}^{K_i} r_{o,t,i} \cdot w_{o,t,i} \right) \sum_{o=1}^{o} R_{o,t}
   \]
   where \( o \) is the type of a resource (considering \( l \) different resource types), \( t \) the timeframe, \( T \) the number of timeframes in the planning horizon, \( r_{o,t,i} \) the type \( o \) resource consumption of project \( i \) in timeframe \( t \), and \( R_{o,t} \) the type \( o \) resource limit in timeframe \( t \). The closer the \( q_3 \) is to one, the better the solution. If a portfolio consumes more resources than are available a simple greedy repair algorithm removes projects until all resource constraints are fulfilled. Then, projects with the worst potential revenue/needed resources ratio are removed first.

4. **Risk** \( (q_4(x)) \). The risk objective expresses a need to minimize a deviation of a median portfolio risk value from a desired risk range. Thus selection pressure towards portfolios with a median risk value in the desired risk range is induced. This objective is calculated as
   \[
   q_4(x) = 1 - d(x)
   \]
   where \( d(x) \) is the percentage deviation of the median portfolio risk value from the desired risk range. The closer the value is to 1, the better the solution. Portfolios with a median risk value in the desired risk range will have their risk objective set to 1. Portfolios with a median risk value not in the desired risk range will be punished having the risk objective value set to a value lower than 1.

5. **Synergy** \( (q_5(x)) \). This quality measure expresses the positive and negative effects among selected projects where the positive synergy is to be maximized while the negative synergy is to be minimized. For this purpose, the sum of negative synergy is subtracted from the sum of positive synergy, consequently referred to as the synergy of a solution. The positive synergy effect is calculated as
   \[
   y^+(x) = \sum_{i=1}^{p} y^+_i(x)
   \]
   , where
   \[
   y^+_i(x) = \sum_{k=1}^{K_i} y^+_{i,k} \cdot w_i \cdot w_k
   \]
   and \( y^+_{i,k} \) is the positive synergy value of project \( i \) concerning project \( k \), \( K_i \) is a set of projects that have a potential synergy with project \( i \), \( w_i \) is 1 if project \( i \) is selected, and 0 if not, \( w_k \) is 1 if project \( k \) is selected, and 0 if not.

Negative synergy effects \( y^- (x) \) are calculated the same way and consequently subtracted from the sum of positive synergy creating the combined maximization synergy objective
   \[
   q_5(x) = y^+(x) - y^-(x).
   \]
   The greater the synergy objective, the better the solution.
Additional objective criteria can be considered as goals to optimize combined with restrictions which define the restrictive characteristic of the criterion. This approach ensures the creation of portfolios which adhere to various preferences, for example number of projects assigned to a certain category, and provide the portfolio manager with a tool to restrict and relax the optimization pressure of these criteria.

4. MULTIOBJECTIVE OPTIMIZATIONS

The presented project selection problem is a typical multiobjective optimization problem where solutions are sought such that they are optimal with respect to all, often conflicting objectives. Typically, no single optimal solution can be found that would be superior to all other solutions with respect to all objectives. Instead, a set of optimal solutions, for which it holds there is no other solution in the whole search space that is superior to these solutions with respect to all objectives, is considered the output of the optimization. A good multiobjective optimization technique must be able to search for a set of optimal solutions concurrently in a single run. This is the reason why EAs become very popular in the domain of multiobjective optimization as they are able to evolve a diverse population of high-quality solutions in parallel.

Perhaps the most widespread and successful are multiobjective evolutionary algorithms (MOEAs) that use a concept of dominance for ranking of solutions. By definition [3], a solution $x$ dominates another solution $y$, if the solution $x$ is no worse than $y$ in all objectives and the solution $x$ is strictly better than $y$ in at least one objective. The concept of dominance can be used to divide any finite set $S$ of solutions into two non-overlapping sets, the non-dominated set $S_1$ and its complement set, the dominated set $S_2$. The set $S_1$ contains all solutions that are not dominated by any other solution of the whole set $S$. The set $S_2$ contains solutions that are dominated by at least one solution of $S_1$. If the set $S$ is the whole feasible search space then the set $S_1$ is a set of optimal solutions called Pareto-optimal solutions and the curve formed by joining these solutions is called a Pareto-optimal front.

In the absence of any higher-level information, all Pareto-optimal solutions are equally important [3]. That is why the goal in a multiobjective optimization is to find a set of solutions that is (i) as close as possible to the Pareto-optimal front and (ii) as diverse as possible so that the solutions are uniformly distributed along the whole Pareto-optimal front.

Examples of successful MOEAs are the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4] and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [20]. They represent state-of-the-art multiobjective optimization evolutionary algorithms. It was shown in [9] that mPOEMS is at least as good as solving the multiobjective 0/1 knapsack problem as the NSGA-II and SPEA2. That is why we decided for adopting the mPOEMS approach for solving the software project portfolio optimization problem since its formulation is very similar to the multiobjective knapsack problem.

5. PROPOSED MPOEMS APPROACH

This section presents the proposed multiobjective optimization approach based on the Prototype Optimization with Evolved Improvement Steps (mPOEMS) [9]. First, a single-objective version of the algorithm is introduced, then its extension for the multiobjective problems and finally the adaptation of mPOEMS to the project selection problem is described.

5.1 POEMS

Prototype Optimization with Evolved Improvement Steps (POEMS) [8] is an iterative optimization approach that employs an EA for finding the best modification of the current solution, called prototype, in each iteration. Modifications are represented as a sequence of fixed length of primitive actions defined specifically for the problem at hand. Besides actions that truly modify the prototype, there is also a special type of action called nop (no operation), which has no effect on the prototype. The evolved action sequences can contain one or more instances of the nop action. Action sequences are assessed based on how well/badly they modify the current prototype, which is passed as an input parameter to the EA. Moreover, sequences that do not change the prototype at all are penalized to avoid a convergence to useless trivial solutions. After the EA finishes, it is checked to determine whether the best evolved sequence improves the current prototype or not. If an improvement is achieved, then the modified prototype is considered a new prototype for the next iteration. Otherwise the current prototype remains unchanged. The iterative process stops after a specified number of iterations. In other words, the POEMS can be considered as an iterative algorithm with evolved hypermutations.

5.2 mPOEMS

mPOEMS is a multiobjective extension of the original single objective POEMS algorithm. Just like NSGA-II and SPEA 2, mPOEMS belongs to the class of multiobjective optimization algorithms that use the concept of dominance. The main differences between mPOEMS and POEMS are:

- mPOEMS maintains a set of best solutions found so far, called a solution base, not just a single prototype solution as in POEMS.
- mPOEMS uses a multiobjective EA based on the dominance concept instead of a simple EA.

The solution base consists of solutions to the problem at hand. mPOEMS improves the solution base in an iterative process, see Algorithm 1. At the beginning of each iteration of mPOEMS one solution from the set of non-dominated solutions of the current solution base is chosen as the current prototype, for which the action sequences will be evolved by the EA. Each MOEA run starts with action sequences
generated at random that are evolved in generations using selection, crossover and mutation steps. New solutions generated by applying the action sequences found by the MOEA to the prototype are merged with the current solution base resulting in a new version of the solution base. Only unique solutions are stored in the solution base. Procedures ChoosePrototype() and Merge() are designed with the aim to keep the set of non-dominated solutions in the solution base as diverse as possible. In ChoosePrototype(), the prototype is chosen among non-dominated solutions in the solution base so that the non-dominated front of the evolved solution base is sampled uniformly. Thus, no “inner” part of the non-dominated front is explicitly preferred. An exception from this rule are the extremal solutions that have the highest probability of being selected, which helps the algorithm to retain the most unique solutions in the solution base. These boundary solutions have in fact the highest probability of being selected, which helps the algorithm to keep the non-dominated front of the evolved solution base wide. Merge() procedure merges solutions produced by action sequences evolved by the MOEA with the current solution base. It uses the crowding distance introduced in [4] in order to retain the most unique solutions in the solution base.

At the end, the set of non-dominated solutions of the final solution base is taken as the result of the run. For the details of the mPOEMS algorithm please see [9].

5.3 Adaptation of mPOEMS to the project selection problem

In this work, each solution is represented by a vector $s = (x_1, x_2, \ldots, x_p)$, where each position $x_i$ describes whether the project $i$ is selected ($x_i > 0$) or not ($x_i = 0$). Value ($x_i > 0$) specifies in which timeframe the project should start.

For the project selection problem the following two actions were introduced:

- **changeMonth(projectId, month)** – This action changes the starting month of a project randomly to another month, with respect to the valid timeframes for the project under consideration. Initialization and the mutation of this action is done randomly, with the restriction that the month value has to be selected, out of the set of valid starting months for the project under consideration.

- **switchOnOff(projectId)** – This action selects a project if it is unselected, and vice versa. It was introduced in order to increase the probability that certain projects will not be selected to the portfolio. Note that projects have usually many more than one starting possibility. Thus, the value zero, which means the project is not selected, has a very low probability of being selected for the changeMonth() action parameter. Initialization and the mutation of this action is done randomly, with the restriction that this action will not be applied to projects which are mandatory.

Simple crossover and mutation operators are used in the MOEA to generate new population of action sequences from the current one. The crossover operator is a uniform-like operator proposed in [9]. Each gene of both parents have the same probability of being selected for the offspring. Valid offspring constitutes an arbitrary combination of parental actions, whereas each gene can be used only once. The mutation operator either changes the action type (changeMonth(), switchOnOff(), nop) or the parameters of an action (projectId or month).

6. EXPERIMENTAL RESULTS

This section presents the experiments carried out to evaluate the performance of the proposed approach.

6.1 COCOMO II Test Set

The test set consists of 50 software projects. The project size predominantly determines the cost of a software project. Source lines of code (SLOC) are used to describe the size. The projects are restricted to having at most 37000 SLOC and at least 1000 SLOC. The maximum duration of a project is restricted to 18 months. The planning horizon is set to three years. The planning horizon is divided into timeframes of the length of a year, resulting in three timeframes. In each timeframe, the available resources are restricted to 500 person-months (PM), resulting in a total of 1500 PM for the planning horizon. This value equals 0.6 percent of the resources needed to conduct all projects in the project pool. Each project has an assigned risk value between 0.20 and 0.80. Since the cost per PM serves as input for the calculation of the potential revenue of a project, this value is set to 5000 Euro per PM. Potential revenue is set to a maximum of 150 percent, and to the minimum of 85 percent of the initial costs. Each project is randomly assigned to exactly one category. The strategic alignment value is calculated following a weighting approach where the alignment value to each strategy was set randomly, resulting in an overall strategic alignment value. A maximum number of thirty percent of all projects is selected to have synergy effects with exactly one project, divided into fifteen percent of the positive synergy and fifteen percent of the negative synergy. Projects which have synergy effects, as well as the projects which have to be selected for the same portfolio in which the synergy effects will take effect, will be selected randomly. Each synergy effect is restricted to be at a maximum of 15 percent of the total cost of the project, which will trigger the synergy effect. A number of 10 percent of all projects is selected randomly to be mandatory and 4 projects are manually selected to be mutually exclusive. The well accepted cost estimation framework COCOMO II (Constructive Cost Model) was used to generate the cost, schedule and effort part of the test set. COCOMO II is a framework for software project cost and schedule estimation, and provides the user with tools for estimating the likely cost, effort and time needed to conduct a software project [2]. Framework parameters were set randomly and are normally distributed. The PM effort per phase was mapped to PM effort per month and costs per month have been calculated.

6.2 Experimental Setup

The mPOEMS was used with the following setting:

- solution base of size 500,
- number of iterations 150,
- length of evolved action sequences 20,
- population of size 50,
• number of generations 20,
• crossover probability 0.75,
• mutation probability 0.15,
• tournament selection with parameter 4.
• the algorithm was run for 150000 fitness function evaluations (i.e. 150000 candidate portfolios generated).

6.3 Performance Measures and Statistical Tests Used for Evaluation

A set of \( k = 20 \) independent runs were carried out and the final non-dominated sets \( NDS_1, \ldots, NDS_k \) of these individual runs were collected. They were then merged in the set \( B = NDS_1 \cup NDS_2 \cup \cdots \cup NDS_k \) and a set \( NDS \) of non-dominated solutions out of that compound set \( B \) was found. A performance of the compared algorithms is assessed based on the quality measures derived from sets \( NDS \), and \( NDS \) achieved by the algorithms. Two performance measures were used in this work:

A hypervolume metric \( H(X) \) also known as size of the space covered, proposed in [19] was used to assess quality of the final set of solutions. A reference volume between the origin and an utopian objective vector (defined as an upper bound value of each objective) is taken into account. Then, the measure is defined as the portion of that reference volume that is dominated by the final set of non-dominated solutions. So, the bigger the value of this measure the better the spread of solutions is, and vice versa.

The \( H(X) \) measure was applied to individual sets \( NDS_1 \). Mean and median values over the set of \( H(NDS_i) \), for \( i = 1, \ldots, k \), were calculated for each algorithm.

6.4 Analysis

We run 20 independent runs with the proposed algorithm. For the performance analysis purposes we used non-dominated solutions (1) from the initial randomly generated solution base and (2) from the final solution base of each run. Out of these individual non-dominated sets we calculated the overall initial non-dominated set \( NDS_{initial} \) and final non-dominated set \( NDS_{final} \). The \( NDS_{initial} \) had 214 solutions whilst the \( NDS_{final} \) had 979 solutions. Note, that in all runs the final solution base contained only non-dominated solutions\(^1\). This difference itself illustrates the ability of the algorithm to evolve high quality solutions (i.e. unique solutions that are not dominated by any other solution generated during the run).

The progress towards the Pareto-optimal front is also well illustrated in terms of the hypervolume metric. First, the utopian objective vector was set to \( U = [1e7, 6.1, 1, 3.5e5] \). Thus, the reference volume between the origin and the utopian vector was 21e12. The randomly generated starting solution base and the final solution base were covering on average 15.2\% (stdev=1.8\%) and 76.4\% (stdev=0.6\%) of the reference objective space volume. Again, this clearly shows the ability of the algorithm to generate solutions close to the Pareto-optimal front. This statement is also supported by plots in 2, where the progress made by the algorithm between the initial and the final solution base is shown. In this figure, a distribution of \( NDS_{initial} \) (red dots) and \( NDS_{final} \) (blue dots) sets is visualized for all 2-objective projections. We can see that in all cases solutions from the \( NDS_{final} \) set are considerably closer to the upper-right corner\(^2\) than solutions from the \( NDS_{initial} \) set.

Another interesting observation is that objectives \( q_1 \) and \( q_2 \) are positively correlated, see 2 b. However, this is not a surprising observation. It is quite natural that expected revenue depends on a proper utilization of resources, thus high revenue can be expected only when resources are effectively used (distributed) throughout the whole planning horizon and vice versa. Generally, observations of this kind are very valuable as they can reveal intrinsic dependencies and relations of the problem at hand.

7. CONCLUSIONS

In this paper a project portfolio optimization strategy is presented and considered as a multiobjective optimization problem. The formulation of the problem at hand takes all of the characteristics relevant to project portfolio optimizing into account, and introduces new and interesting concepts, e.g. optimizing resource usage over timeframes.

The authors propose the usage of the evolutionary-based multiobjective algorithm mPOEMS for solving the problem at hand. The approach proved to work efficiently and it is shown in this paper that the presented approach is capable to optimize the given test set for the given portfolio optimization criteria. Most valuable insights are that the complex project portfolio optimization problem with its various constraints and objectives can be solved with the presented approach and that the presented strategy performs highly efficient in regard of the optimization of each of the considered objectives and all of their combinations.

From the practical point of view it is very important and useful that the project portfolio is optimized with respect to all of the optimization criteria simultaneously providing a user with a set of high-quality solutions from which he/she can choose the best one according to his/her posterior priorities in the end.

The choice for the mPOEMS algorithm was made based on the fact that mPOEMS has been shown to outperform the state-of-the-art multiobjective evolutionary algorithms NSGA-II and SPEA2 on the multiobjective 0/1 knapsack problem [9], whose formal specification is very similar to the one of the project portfolio optimization problem. However, further research will focus on investigation of the efficacy of the presented approach and comparison analysis of the mPOEMS with NSGA-II and SPEA2.

8. REFERENCES


\(^{1}\)That is obviously not true for the randomly generated starting solution base.
\(^{2}\)Note, since all of the objectives are to be maximized the upper-right corner of the 2D projection of the objective space represents the region of interest.
Figure 2: Illustration of an improvement of the final solution base (blue dots) over the initial solution base (red dots). For the sake of easy visualization 2D projections of the two sets of non-dominated solutions for all pairs of objectives are used.


