Multiobjective Optimization for Project Portfolio Selection

Márcio de O. Barros[1], Hélio R. Costa[2, 3], Fábio V. Figueiredo[1], Ana Regina C. da Rocha[2]

[1]Post-graduate Information Systems Program – PPGI/UNIRIO
Av. Pasteur 458, Urca – Rio de Janeiro, RJ – Brazil

[2]COPPE / UFRJ – System Engineering and Computer Science Department
Caixa Postal: 68511 – CEP: 21945-970 – Rio de Janeiro – Brazil.

[3]CCA-RJ – Brazilian Air Force, Ponta do Galeão s/nº - Ilha do Governador
CEP: 21941-510 – Rio de Janeiro – Brazil

{marcio.barros,fabio.figueiredo}@uniriotec.br, heliorcosta@gmail.com, darocha@centroin.com.br

ABSTRACT
This paper proposes a multiobjective heuristic search approach to support a project portfolio selection technique on scenarios with a large number of candidate projects. The original formulation for the technique requires analyzing all combinations of candidate projects, which is unfeasible when more than a few alternatives are available. We have used a multiobjective genetic algorithm to partially explore the search space of project combinations and select the most effective ones. We present an experimental study based on four project selection problems that compares the results found by the genetic algorithm to those yielded by a non-systematic search procedure. Results show evidence that the project selection technique can be used in large-scale scenarios and that GA presents better results than simpler search strategy.

Categories and Subject Descriptors
D.2.9 Management

General Terms
Algorithms; Project Management; Risk; Experimentation.

Keywords
Multiobjective optimization; portfolio selection; risk; dependency.

1. INTRODUCTION
ACM 978-1-4503-1178-6/12/07.Selecting the projects which will be executed by a company is a major component of portfolio management [1]. Project selection aims to define an (close to) optimal subset of projects to comprise the company's portfolio, taking into account their characteristics and relationships. Many project selection techniques can be found in the literature [1] [2], but few of them address an aspect that becomes important if these projects are to be executed and managed together: the dependencies among candidate projects.

Recently, Costa et al. [3] presented a project selection technique based on Modern Portfolio Theory [4]. The technique evaluates all portfolios which can be formed by combining a set of candidate projects, introduces a systematic procedure to calculate the dependencies among them, estimates the risks of all portfolios prone to be selected, and generates a return x risk indicator for each portfolio. However, the cost of executing the technique is a power function of both the number of candidate projects and the number of risks that may affect these projects. The high cost is due to analyzing all combinations of projects and prevents using the technique in large scenarios, with more than a few candidates.

In this paper we present a multiobjective heuristic optimization approach to support the application of the technique proposed by Costa et al. [3] in large-scale scenarios on regard of the number of candidate projects available to comprise the portfolio. We use a bi-objective GA to find effective portfolios in terms of their return x risk profiles without examining all possible combinations of the available projects. The optimization approach was empirically evaluated and results show that the heuristic search can find efficient portfolios in feasible time and yields better results than a simpler search procedure.

2. PROJECT PORTFOLIO SELECTION AS A MULTIOBJECTIVE PROBLEM
Project portfolio selection is a bi-objective problem where two incomparable measures (risk and return) define the most effective portfolios. Risk must be minimized, while expected return must be maximized. Therefore, we are interested in the portfolio which yields maximum return for a given level of risk or, on the opposite perspective, which incurs minimum risk to yield a certain return. The most effective portfolios form a curve disposed in the risk x return plane. A decision about which among these portfolios will be undertaken by the company depends on the decision-makers willingness to accept more risk in exchange for more return.

A bi-objective search to select the most effective portfolios must look for the Pareto-optimal set of subsets of candidate projects maximizing return and minimizing risk. We have addressed this optimization problem using the NSGA-II algorithm [5] and devised a systematic approach to select its parameters. The algorithm was configured to use single point crossover with 90% crossover probability and uniform mutation with 1% probability. Binary tournament is used as selection strategy. Population size was set as twice the number of projects. The maximum number of fitness function evaluations was set as 100 times the square of the number of projects. Each candidate solution represents a potential portfolio and was encoded as a sequence of bits, one for each available candidate project. The bit for a given project indicates whether the project is part of the portfolio represented in the solution.

3. EVALUATING THE APPROACH
We have analyzed the behavior of the NSGA-II algorithm using four real-world instances. These instances were provided by a

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

Brazilian company acting in the distribution of electric energy and depict an excerpt of the candidate projects that were available to form the company's project portfolio for 2011. They contained, respectively, 25, 50, 75, and 100 projects. Each instance had 10 risks described by their probability of occurrence and financial impact upon each project comprising the instance. Finally, each instance specified a limit upon the amount of capital available to be invested on its projects.

To evaluate whether a complex search procedure, such as the NSGA-II algorithm, would be required to find good solutions for the project selection problem in scenarios of varying sizes, we have designed and executed an experimental study to compare the heuristic search with a simpler, non-systematic search procedure. Two configurations were tested for each instance. The first one (GA) used the NSGA-II algorithm with the parameter settings and fitness evaluation budget described in Section 2. The second configuration (RS) used a multiobjective random search with the same fitness evaluation budget given to the NSGA-II algorithm.

Each configuration was executed 30 times for all instances. For each pair of configuration and instance, each running cycle yielded a Pareto front comprised of a finite set of solutions (PF_i). After running all cycles for a given instance and configuration, a best front for that pair was built by joining the fronts yielded by each cycle and removing dominated solutions (PF_{GA} and PF_{RS}). Finally, PF_{GA} and PF_{RS} were merged to create the best front for the instance at hand (PF_{best}), again removing dominated solutions. Each vertex of the Pareto fronts represents a portfolio and is described by its expected return and its risk.

To evaluate the efficiency of a configuration, we have collected the execution time for each cycle, configuration, and instance. In this context, execution time means the wall-clock time required to run the cycle. To evaluate the effectiveness of a configuration, we collected the error ratio indicator for each cycle, configuration, and instance. Configurations were compared in a per instance basis, e.g., results yielded by GA for the instance with 25 projects were compared to those presented by RS for the same instance. Lower execution times for a given configuration indicate that it is more efficient than the other. Smaller error ratios for a given configuration denote that it yields more effective results than the second one. These values were subjected to a non-parametric Wilcoxon-Mann-Whitney test to ascertain if there was statistically significant difference between the configurations.

The following tables present means and standard deviations of the measures above for each instance/configuration over 30 cycles. They also present the *p-value* for the non-parametric test. Table 1 shows execution times (measured in seconds) collected after performing the experiment. Execution time for configuration GA is on average two times greater than under configuration RS, but this percentile is substantially reduced for the largest instance. Nevertheless, NSGA-II consumes much more processing time than the random search to find its solutions. The *p-value* for the statistical test converges to zero for all instances, denoting that differences in execution time are significantly different with, at least, 99% confidence.

Table 2 shows error ratios collected after running the experiment. As in the former Table, it presents means and standard deviations for each instance's error ratio under configurations GA and RS over the 30 cycles and the *p-value* for the statistical test. Error ratio under configuration RS is, on average, 98% greater than under GA. For all but the smallest instance, no cycle running random search contributed to PF_{best}. Since smaller values are preferred, the genetic algorithm seems to find more effective solutions (in terms of error ratio) than random search. As in the former Table, *p-values* converge to zero for all instances, denoting that differences in error ratio are statistically significant with at least 99% confidence.

Table 1 – Execution time analysis

	GA	RS	P-Value
25P	1.7 ± 0.05	1.3 ± 0.19	< 0.001
50P	13.4 ± 0.03	5.2 ± 0.73	< 0.001
75P	46.1 ± 0.03	7.2 ± 0.23	< 0.001
100P	134.7 ±0.27	80.0 ± 12.7	< 0.001

Table 2 – Error ratio analysis

	GA	RS	P-Value
25P	0.45 ± 0.07	0.90 ± 0.03	< 0.001
50P	0.53 ± 0.06	1.0 ± 0.0	< 0.001
75P	0.41 ± 0.04	1.0 ± 0.0	< 0.001
100P	0.63 ± 0.05	1.0 ± 0.0	< 0.001

4. CONCLUSIONS

Data presented in Tables 1 and 2 shows sound evidence in favor of the heuristic search, except for small instances with relatively large budgets to fund the project portfolio. NSGA-II outperformed random search in finding solutions closer to the best Pareto front. On the other hand, random search seems a feasible alternative for small instances or those with large budgets. Limitations of the present work include adapting the heuristic search to deal with a large number of risks and repeating the experiment with more instances and different algorithms.

ACKNOWLEDGMENTS

The author would like to express his gratitude for FAPERJ and CNPq, the research agencies that supported this project.

REFERENCES

[1] Killen, C.P., Hunt, R.A., Kleinschmidt, E.J., "Managing the New Product Development Project Portfolio: A Review of the Literature and Empirical Evidence". PICMET Proceedings, Portland, USA, August 2007.

[2] Cooper, R.G., Edgett, S.J., Kleinschmidt, E.J., *Portfolio Management for New Products*. 2nd edition, Cambridge, MS, USA, 2001.

[3] Costa, H.R., Barros, M.O., Rocha, A.R.C, "Software Project Portfolio Selection: a Modern Portfolio Theory Based Technique", Proceedings of the 22nd Software Engineering and Knowledge Engineering Conference, San Francisco, EUA, 2010.

[4] Markowitz, H.M., "Portfolio Selection". Journal of Finance, Vol. 7, No 1, pp. 77-91, 1952.

[5] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computing Vol. 6, No. 2, pp 182-197, April 2002.