Evaluating and Selecting Optimization Software Packages: A Framework for Business Applications

Waleed Abohamad, Amr Arisha

Abstract—Owing the fact that optimization of business process is a crucial requirement to navigate, survive and even thrive in today's volatile business environment, this paper presents a framework for selecting a best-fit optimization package for solving complex business problems. Complexity level of the problem and/or using incorrect optimization software can lead to biased solutions of the optimization problem. Accordingly, the proposed framework identifies a number of relevant factors (e.g. decision variables, objective functions, and modeling approach) to be considered during the evaluation and selection process. Application domain, problem specifications, and available accredited optimization approaches are also to be regarded. A recommendation of one or two optimization software is the output of the framework which is believed to provide the best results of the underlying problem. In addition to a set of guidelines and recommendations on how managers can conduct an effective optimization exercise is discussed.

Keywords—Complex Business Problems, Optimization, Selection Criteria, Software Evaluation.

I. INTRODUCTION

R APID advances in information technology, global outsourcing, and diversity in products and services have a substantial effect on the acceleration of dynamics in business systems (e.g., manufacturing systems, healthcare systems ...etc). Understanding and improving the performance of such systems is a complex and challenging task which is imputable to high level of uncertainty, conflicting objectives, lack of necessary information, large number of constraints, and immense number of inter-connected components and decision variables. Making decisions that lead to the optimum performance of these systems seems to be impossible. With such level of complexity in the business environment, optimization has a potential to make a significant contribution to resolve these challenges.

The increase of the available optimization software packages has resulted in a selection dilemma to the business managers because of various prices, many features and the diverse compatibility of every commercial package. Thus, business managers face numerous difficulties for selecting an optimization package that best suits their needs and budget.

W. Abohamad is with the school of Management, Dublin Institute of Technology (DIT), Aungier St., Dublin 2, Dublin, Ireland (phone: +353(0)87-165-3983; e-mail: waleed.abohamad@dit.ie).

A. Arisha is with the school of Management, Dublin Institute of Technology (DIT), Aungier St., Dublin 2, Dublin, Ireland (e-mail: amr.arisha@dit.ie).

Improper selection of an optimization package may result in wrong strategic decisions with subsequent economic loss to the organization [1]. Evaluation and selection of an optimization package is a complex process in which individual packages are not evaluated in isolation, but in the context of intended use. Technical criteria are insufficient for determining the suitability of the software; other concerns have to be considered such as vendor reputation, end users experience, and a wide range of other criteria [2]. Software packages often provide features and functions that overlap with other existing packages in the organization. Identifying and considering these interactions is another added complexity to the evaluation process. Another difficulty associated with the selection process is the rapid technology advances which affect the viability of software packages. Other non-technical difficulties arise when the involved stakeholders within the company suffer a conflict of interests. Thus, the evaluation and selection process has to deal with this inevitable situation to aid in resolving these cross-purposes. Therefore, selecting an optimization tool that meets the requirements requires a full examination of many conflicting factors. In this paper, a framework is proposed for evaluation and selection of optimization software.

II. SOFTWARE SELECTION METHODOLOGY

Preceding the evaluation process, criteria for the selection of optimization software packages have to be considered. Identifying these criteria is a key for a successful evaluation process. Many researchers have provided a list of criteria for software packages evaluation relating to specific fields such as simulation [3]-[4], customer relationship management (CRM) [5], enterprise resource planning (ERP) [6], and knowledge management (KM) [7]. Unfortunately there is no generic list of criteria that can be used for evaluating any of these software packages [1]. Technology advances, changes in packages features, and various terminologies used by experts are behind the lack of such a standard [8]. Although, functional criteria for software selection may vary from subject to subject and from package to package, criteria related to vendor, cost, and quality of the software may be common and can be used for selection of any software package [9]. But, specific criteria for optimization packages have not been investigated by researchers. A detailed knowledge of the selection criteria and the score of the alternatives on these selection criteria is required for the

evaluation and selection. Moreover, the process of evaluation becomes more challenging with the existence of many alternatives and a longer criteria list. As a result, evaluation and selection of an optimization package is a time-consuming task unless an efficient methodology is used. Basically, a selection methodology points up the issues and factors that should be considered during the evaluation and selection of software packages. The methodology is intended as a guideline that can be adapted according to the requirements of individual organization; it is not intended as a rigid structure that must be followed without any deviation [10]. A generic stage-based methodology is proposed by [1] in their recent review on evaluation and selection of software packages. Seven stages constitute the methodology:

1) Determining the needs and requirements for purchasing the software package.
2) Searching for software packages that might be a potential candidate, including an investigation of software characteristics and capabilities provided by vendor.
3) Screening of candidate packages by excluding packages that do not have the required features or are not compatible with existing hardware and software.
4) Evaluating and ranking the remaining packages.
5) Obtaining a demo version of top software packages and conducting an empirical testing.
6) Negotiating the software vendor about the software price, number of licenses, functional specification, and maintenance responsibilities.
7) Purchasing and implementing the most appropriate software package.

In order to support these activities, a tool or a support system is needed. Table I provides a summary of research work done in developing systems/tools for evaluating and selecting software packages. The table highlights the degree to which each stage of evaluation and selection methodology is included.

TABLE I
SELECTION METHODOLOGY STAGES AND ASSOCIATED APPLICATION DOMAIN IN RELATED LITERATURE

		Reference					
		[4]	[5]	[6]	[10]	[11]	[12]
Methodology Stage	Stage (1)		x		x	x	
	Stage (2)		x		x	x	
	Stage (3)	x		x			x
	Stage (4)					x	x
	Stage (5)			x	x		
	Stage (6)		x	x			
Application Domain [a]		Sim	CRM	ERP	KM	ESS	COTS

[a] Sim = Simulation; CRM = Customer Relationship Management; ERP = Enterprise Resource Planning; KM = Knowledge Management; ESS = Expert Systems Shells; COTS = Commercial Off-The Shelf.

It is evident that none of these frameworks support all the stages of the selection methodology. This is because some of these activities are directly related to the organization or company, such as determining the needs and requirements, negotiating software vendors, and purchasing the software. On the other hand, other activities and stages consume more time and require more technical skills and consulting domain experts [13]. In addition, there is no investigation of frameworks or tools for selecting optimization software package for business applications. Given that, this paper presents a comprehensive framework developed to include a software selection methodology, an evaluation technique, evaluation criteria, and a system to support evaluation and selection of optimization software packages.
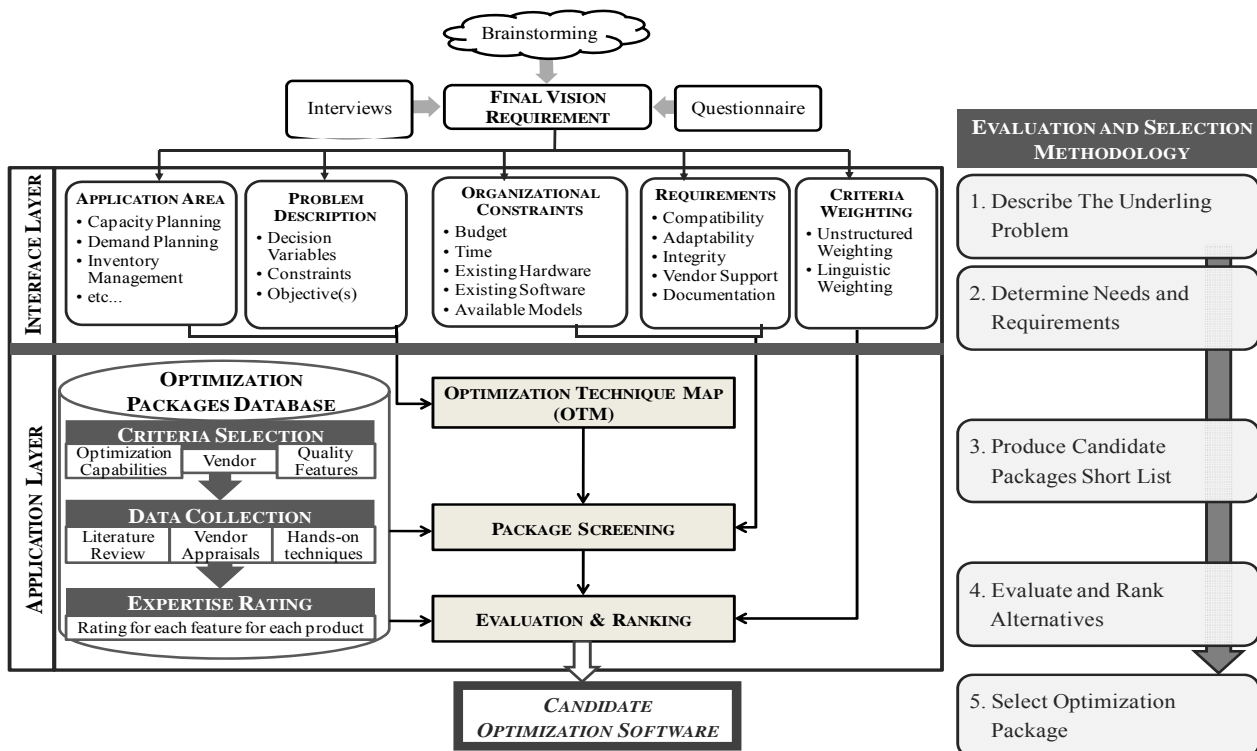
III. SELECTION AND EVALUATION FRAMEWORK FOR OPTIMIZATION PACKAGES

A. Framework Overview

Time-consuming activities such as searching for software packages, selecting criteria, data collection, software screening, and software evaluation are the core of the framework (Fig. 1). The availability of large number of software packages and new releases of packages with new or modified features is a major challenge. We overcome this challenge by building a database for optimization software packages which is continuously updated. The database stores information about software packages and their features. The features and characteristics of software have to be carefully chosen to reflect the products capabilities and meanwhile to show other qualitative features such as software reliability and performance. These criteria have been selected and designed in a hierarchical structure considering functional, quality, and vendor characteristics. Some of these criteria are incompatible such as licence cost, vendor reputation, and functionality features. Consequently, Analytical Hierarchical Process (AHP) [14] is used to deal with this multi-criteria decision making situation. Additionally, AHP considers different importance and weights for criteria from the point of view of company stockholders.

B. Framework Architecture

The organization forms a management team to identify both business needs and requirements [15]. The task of the team is mainly to determine the needs and reasons for applying optimization software. For example, the need to optimism staff schedules or to have an optimal inventory level. Afterwards, a further description about such needs have to be detailed in terms of decision variables to be optimized, constraints, and desired objectives to be achieved. Furthermore, technical considerations related to the desired optimization software should be considered and specified at this level. Examples of such considerations might include compatibility with other products in use, adaptability, flexibility, reliability, maintainability, impact on system integrity and integration, vendor support, training, documentation, and licenses.

Fig. 1 The proposed framework for optimization software selection

Finally, organizational constraints should be determined and considered in the selection process such as budgetary constraints available for purchasing the tool, the time frame for completing the installation and implementation of the package, and establishing what hardware and software currently exists to make sure that the optimization software is compatible. Accordingly, brainstorming, interviews, and questionnaires are used to carry out these tasks to extract the necessary information.

1. Interface Layer

The interface layer of the proposed framework organizes the final requirements and needs of the organization management team into five main groups:

a. Application Area: Such as capacity planning, demand planning, inventory management, etc.
b. Problem Description: Decision variable, constraints, and objectives.
c. Organizational Constraints: Budget, time, existing hardware and software, and available system models.
d. Technical Requirements: Such as compatibility, adaptability, vendor support, etc.
e. Criteria Weighting: Weighting of desired features by management team.

2. Application Layer

The application layer consists of four main components: a map of optimization techniques, a database of optimization packages, a screening mechanism, and evaluation and ranking approach. As its input, the Optimization Techniques Map (OTM) - described in section IV - uses the application area and problem description to identify the optimization technique that can contribute effectively in solving the optimization problems of the organization. Potential packages are then selected for evaluation from the optimization software database in a screening process which considers the suitable optimization technique and the technical requirements of the organization. The optimization packages database is developed using specific criteria - described in section V - that are usually used to evaluate different optimization software packages in the market place. After the screening process, the potential packages are evaluated using AHP. Because of the hierarchical structure of the proposed evaluation criteria, AHP consolidates information about alternative software packages effectively using multiple criteria. Subsequently, the results of the evaluation are presented in which the package with the highest rank is recommended as the best selection.

IV. OPTIMIZATION TECHNIQUES MAP - OTM

Choosing incorrectly the optimization technique required for the business application can consequently lead to a wrong selection of the optimization software package. Thus, business will face many biased decisions as well as incorrect strategic directions that mostly cause financial losses. The suitability of optimization technique depends on the type of application area, the nature of decision variables and constraints, and the

target objectives from the optimization process. Many classification schemes have been proposed in literature for classifying optimization techniques. For example, decision variables can be used to classify optimization methods into continuous input parameter methods and discrete input parameter methods [16]. Continuous input parameter methods include gradient and non-gradient methods, on the other hand; discrete input parameter methods include statistical methods, ordinal optimization, and meta-heuristics algorithms. The shape of the response surface (i.e. global as compared with local optimization) can be used also to categorize optimization techniques into local optimization techniques (e.g. discrete decision space and continuous decision space) and global optimization techniques (e.g. meta-heuristic, sampling algorithms, and gradient surface method) [17]. Unfortunately, these classifications do not consider the fact that real-world business systems have a critical feature of multiple conflicting objectives. Optimization techniques can also be classified based on the timing of when the preference information of decision maker or system optimizer is gathered: progressive, apriori, posterior or no articulation of preference [18]. One of the main factors that control the way methods are classified is the measurement tool that is used to assess the system performance (e.g. simulation models and/or mathematical models). Simulation models are only considered in the aforementioned classifications which neglect other modeling approaches. System modeling methods and different approaches of uncertainty handling have to be considered to provide a consistent and comprehensive classification of optimization methods [19].

A major challenge that faces researchers and managers is the lack of a clear guideline and comprehensive classification that considers different aspects of the underlying problem. In [20], optimization mechanism, decision variables, and dimension of the solution space have been concurrently considered in selecting optimization technique for supply chain application. This paper extends that study to include the modeling approach: mathematical models and simulation models. Showing in Fig. 2, is the Optimization Techniques Map (OTM). From left to right, the OTM starts with the modeling approach to classify the optimization techniques into mathematical programming and direct search methods. Afterwards, decision variables and solution space are used respectively for further categorization. The OTM can be viewed at the right side as a classification of methods in terms of the optimization mechanism: mathematical programming, gradient-based, meta-model-based, statistical methods, and meta-heuristics.

A. Mathematical Programming

Mathematical programming methods are used for problems that can be modeled using equations to describe system constraint(s) and objective function(s). If a problem can be described using certain sets of equations, then an optimum solution can be computed following a prescribed algorithm or technique.

B. Gradient-based Methods

Differentiation, in the gradient context, is usually used to simplify the objective function in order to find an optimum solution. The Gradient-based approach is subject to have a mathematical expression of the objective function. When such a mathematical expression cannot be obtained, there is a requirement for an estimation technique to start the solution procedure. The estimated gradients direction guides the search process to move from one potential solution to another in an iterative scheme in a process known as stochastic approximation (SA). During the iteration phase, the step size is controlled by the gradient estimator which is embedded in optimization algorithms.

C. Meta-model-based Methods

While gradient-based estimators are used to estimate the derivatives of the objective function, meta-model-based techniques use analytical approach to approximate the objective function itself. The analytical model is developed based on the relationship between the decision variables (i.e., input) and the simulation model output. The meta-model can replace a part of the simulation model with a mathematical function that mimics the input-output behavior of that part, with respect to some measure of interest to the system analyst.

D. Statistical Methods

Gradient-based and meta-model-based optimization techniques are used in case of continuous parameters. In discrete decision parameters, the problem is to select one of the predetermined system configurations (i.e., there is a finite solution space). The task of optimization algorithms is to select one of these configurations that achieve the best performance of the system based on the selected criteria. Because the system performance is not deterministic and so too the output of a stochastic simulation model, further statistical analysis is required to compare alternatives.

E. Meta-heuristics

Statistical methods were successfully used in problems with discrete decision parameters. But, it is computationally infeasible to evaluate every possible alternative or all parameter combinations when the solution space is very large. Meta-heuristic algorithms are used in such instances to efficiently guide the search process towards potential solutions. They ultimately balance between exploration of solution space and exploitation of good solution(s) to overcome the conflict between local optimum solutions and the global ones. This is performed in an iterative process by initially starting with a solution (point-based) or set of solutions (set-based or population based). Then in each iteration the search progresses to a new, possibly better, solution(s) in the neighborhood of the current solution. Each meta-heuristic method has its own mechanism to define the neighborhood structure.
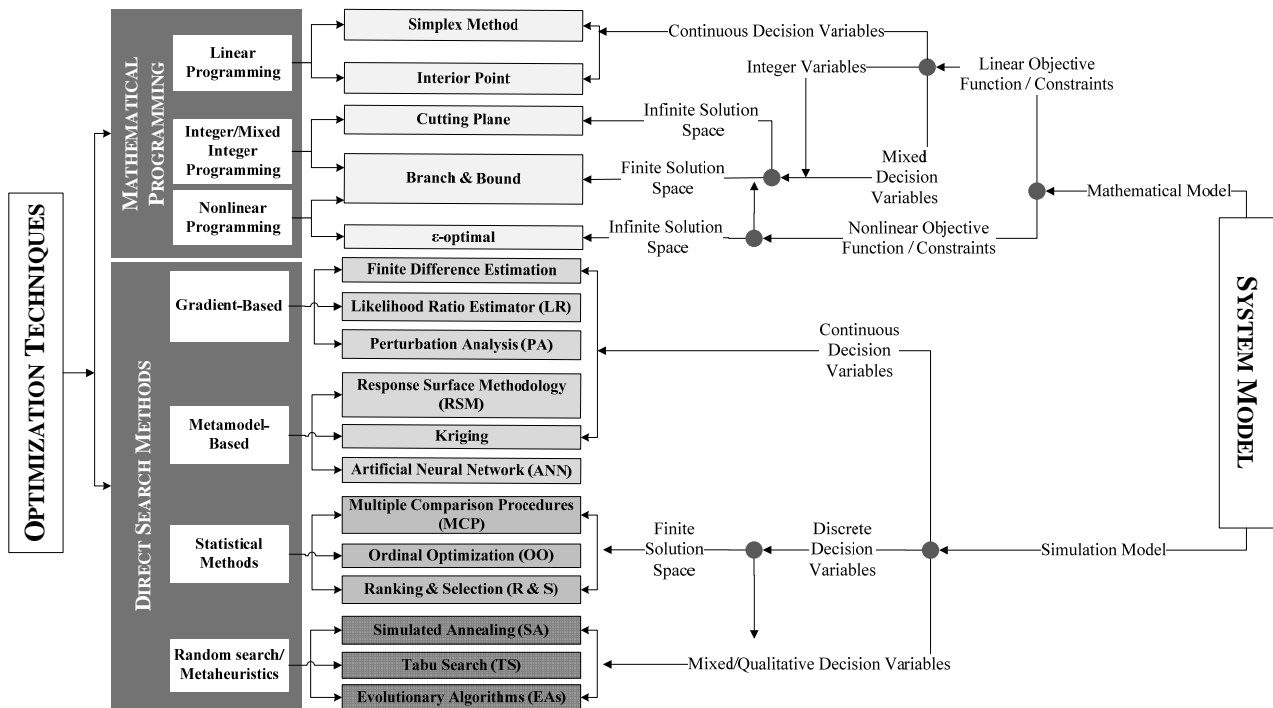
Fig. 2 Optimization Techniques Map (OTM)

V. EVALUATION CRITERIA FOR OPTIMIZATION SOFTWARE

Selection of optimization software is considered as being a multi-criteria problem. Having most of the optimization software with many features, this study has only identified six sub-criteria for quality features as: adaptability, compatibility, portability, scalability & hardware, reliability, and usability. Besides the optimization capabilities and quality features of the software, vendor reputation and software cost are important elements which should be considered in the selection process. These three main criteria represent the highest level of the hierarchy; optimization capabilities, quality features, and vendor.

A. Optimization Capabilities

Criteria in this group relate to the optimization capabilities supported by the software. They include the number and type of decision variables that the software supports, the number and type of constraints, the optimization approach used by the package, and the multi-objective approach used by the software package to manipulate the multi-objective nature of most complex business systems.

1. Decision Variables

Decision variables can take a discrete set of values such as the number of servers in the system, alternative locations of depots, different scheduling rules or policies, etc. On the other hand, in a continuous space, the feasible region consists of real-valued decision variables such as order quantity and reorder quantity in inventory problems, release time of factory orders. Decision variables can be qualitative (e.g., queuing strategies) or a mix between discrete and continuous values. Optimization methods differ in the way they can handle these situations.

2. Objective Functions

Criteria in this group include the type of approach used to manage the multi-objectives element. Most real-world problems are inherently characterized by multiple and conflicting objectives. Approaches for multi-objective optimization can be classified in two general categories: aggregate-based approach and Pareto-based approach.

a. Aggregate-based Approach: is used to combine objectives into a single objective function by adapting one of the following:
- Weighted sum approach: This approach transforms the multi-objective optimization problem into a scalar one by adding all the objectives together using different coefficients for each objective.
- Goal programming: In this method, each objective is assigned targets (goals) by the decision maker that wishes to achieve. These values are incorporated into the optimization problem as additional constraints. The objective function will then try to minimize the absolute deviations from the targets to the objectives.
- ε-Constraints approach: The ε-Constraints approach is based on optimizing one objective function (selected by the decision maker), considering the other objectives as constraints bound by some allowable level ε.

b. Pareto-based Approach: This approach considers all the objective functions concurrently. It is rare to find a common solution that has an optimum value for all the objective functions. Accordingly, a set of solutions called "Pareto optimal solutions" is produced.

3. Optimization Algorithm Approach

Criteria within this group discuss the optimization approach implemented by the software, as discussed in section IV, optimization approach can be: mathematical programming, gradient-based, meta-mode-based, statistical, and meta-heuristics.

B. Quality Features

The quality model approach suggested by [21] is used to identify the quality features of optimization software packages which are then classified into six groups; adaptability, compatibility, portability, maintainability, reliability, and usability.

1. Adaptability

Adaptability refers to the level of customization in software package. For instance, checking whether the software permits the users to personalize layout, reports, and modules or not. Mostly, this is directly related to coding aspects of the software. Criteria in this group have been classified into customization and coding aspects. A summary of explanations of each sub-criterion is given in Table II.

TABLE II
CRITERIA RELATED TO SOFTWARE ADAPTABILITY

Criteria Group	Criteria	Explanation
Customization	Layout	Ability to personalize the layout of package interface
	Reports	Ability to personalize the layout of reports produced by package
	Modules	Ability to personalize modules by programming languages
Coding Aspects	Programming	Availability of programming capabilities
	Link to Other Languages	Possibility to link the package to other programming languages
	Access to Source Code	Accessibility level of users to the source code
	Code Generator	Ability of the package to provide a program code for the optimization model, which is possible to be modified
	Functions	Availability of a library of in-built functions and the possibility of defining functions by user

2. Compatibility

These criteria evaluate whether the package can be integrated with other software packages such as spreadsheet packages, statistical packages, database management systems, simulation packages, ERP systems, and other optimization packages.

3. Portability

This feature allows the optimization tool to run on a wide range of platforms and to support different data exchange standards such as EDI and XML.

4. Scalability and Hardware

Scalability is a desirable feature of optimization software as it indicates its ability to manage an increasing number of workload and users. Communication protocol and number of workstations contribute to a great extend to the scalability feature of the software package (Table III).

TABLE III
HARDWARE AND SOFTWARE SCALABILITY CRITERIA

Criteria Group	Criteria	Meaning
Scalability	Number of Modules	Average size of independent code units
	Level of Independence	Number of independently installable modules
	Number of simultaneous Users	Number of simultaneous users that can be linked and served by the system
Hardware	Communication Protocols	Communication protocols supported by the package
	Internal/External Memory	Primary/secondary storage needed to run the software package
	Number of Workstations	Maximum number of users that can be supported

5. Reliability

Criteria classified in this group measures the capability of the software package to run consistently without problems, the capability of the software package to support backup and recovery feature, and the security levels (e.g. user identification, auditing, and data encryption).

6. Usability:

Usability can be defined on three prospects:
- User experience level: Ability of the software package to support beginners, intermediate, and advanced users or a combination of user types.
- Data visualization: Capability of the software package to present data effectively
- Output: Standard and customized report facilities of the software package, output to file printer, plotter etc.

C. Vendor

This group considers criteria related to the evaluation of the credibility of the vendor, and the optimization software package. The cost of the software including the licence price, installation cost and implementation cost, the training cost, and the upgrading cost are very important criteria for evaluation (Table IV).

VI. CONCLUSION

The growing complexity of managing business along with the need for a more efficient performance, greater agility, better product quality and lower cost urged the use of optimization techniques in business applications.

TABLE IV
VENDOR RELATED CRITERIA

Criteria Group	Criteria	Criteria Meaning
Vendor & Product Information	Pedigree	History and popularity of vendor and product in market
	Documentation	Availability of user manual and tutorials to learn how to use the software package
	User Support	Availability of training courses to learn the package
Cost	Licence cost	Licence cost of the product in terms of number of users
	Training cost	Cost of training to the users of the system
	Installation & Implementation cost	Cost of installation and implementation of the product
	Upgrading cost	Cost of upgrading of the product when new version will be launched

The challenge of selecting appropriate optimization software has risen in importance because of the increased number of optimization software packages on the market. The paper addresses a list of the most important criteria, which reflect the issues that should be considered in evaluating and selecting the optimization software package. The comprehensive manner of classification of the selection criteria provide an easy guide to the user when evaluating and selecting the optimization software package. Considering the company requirements and needs, the framework suggests the best fit software. The friendly user interface of the framework makes it easy to be used by non-expert users in various organizations with different business needs and complexities. The ability of end users to assign weights to the criteria increases the reliability of the framework and helps to cope with changes in the business requirements. Integrating an outstanding structured database with the selection criteria has significantly reduced the cycle time required for the overall selection and evaluation process. Finally, the study shows a potential of web database development that might help to standardize the criteria terminology while facilitating common measures for the selection process.

REFERENCES

[1] A. Jadhav and R. Sonar, "Evaluating and selecting software packages: A review," *Inform. Software Tech.*, vol. 51, no. 3, pp. 555–563, 2009.
[2] S. Comella-Dorda, J. Dean, E. Morris, and P. Oberndorf, "A process for cots software product evaluation," in *COTS-Based Software Systems,* ser. Lecture Notes in Computer Science, J. Dean and A. Gravel, Eds. Springer Berlin / Heidelberg, 2002, vol. 2255, pp. 86–96.
[3] J. Nikoukaran, V. Hlupic, and R. Paul, "A hierarchical framework for evaluating simulation software," *Simulat. Pract. Theory*, vol. 7, no. 3, pp. 219–231, 1999.
[4] T. W. Tewoldeberhan, A. Verbraeck, E. Valentin, and G. Bardonnet, "Software evaluation and selection: An evaluation and selection methodology for discrete-event simulation software," in *Proc. 34th conf. Winter Simulation*, San Diego, California, 2002, pp. 67–75.
[5] E. Colombo and C. Francalanci, "Selecting CRM packages based on architectural, functional, and cost requirements: Empirical validation of a hierarchical ranking model," *Requirements Engineering*, vol. 9, no. 3, pp. 186–203, 2004.
[6] X. B. Illa, X. Franch, and J. A. Pastor, "Formalising ERP selection criteria," in *Proc. 10th Intl. Workshop on Software Specification and Design*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 115–122.
[7] E. Ngai and E. Chan, "Evaluation of knowledge management tools using AHP," *Expert Syst*. Appl., vol. 29, no. 4, pp. 889–899, 2005.
[8] J. Nikoukaran and R. Paul, "Software selection for simulation in manufacturing: A review," *Simulat. Pract. Theory*, vol. 7, no. 1, pp. 1–14, 1999.
[9] A. Arisha and M. El Baradie, "On the selection of simulation software for manufacturing application," in *Proc. 19th Intl. Manufacturing Conf.*, Belfast, N. Ireland, 2002, pp. 495–507.
[10] N. Patel and V. Hlupic, "A methodology for the selection of knowledge management (KM) tools," in *Proc. 24th Intl. Conf. Information Technology Interfaces*, Cavtat, Croatia, 2002, pp. 369–374.
[11] A. Stylianou, G. Madey, and R. Smith, "Selection criteria for expert system shells: a socio-technical framework," *Commun. ACM*, vol. 35, no. 10, pp. 30–48, 1992.
[12] J. Bhuta and B. Boehm, "A method for compatible cots component selection," in *COTS-Based Software Systems*, ser. Lecture Notes in Computer Science, X. Franch and D. Port, Eds. Springer Berlin / Heidelberg, 2005, vol. 3412, pp. 132–143.
[13] J. Kontio, "OTSO: A systematic process for reusable software component selection," Univ. of Maryland Institute for Advanced Computer Studies Report No. UMIACS-TR-95-63, College Park, MD, USA, Tech. Rep., 1995.
[14] T. Saaty, "How to make a decision: The analytic hierarchy process," *Eur. J. Oper. Res.*, vol. 48, no. 1, pp. 9–26, 1990.
[15] J. Curry, Jr. and D. M. Bonner, *How to Find and Buy Good Software: A Guide for Business and Professional People.* Prentice Hall Professional Technical Reference, 1983.
[16] J. Swisher, P. Hyden, S. Jacobson, and L. Schruben, "Simulation optimization: A survey of simulation optimization techniques and procedures," in *Proc. 32nd conf. Winter Simulation*, San Diego, CA, USA, 2000, pp. 119–128.
[17] E. Tekin and I. Sabuncuoglu, "Simulation optimization: A comprehensive review on theory and applications," *IIE Trans.*, vol. 36, no. 11, pp. 1067–1081, 2004.
[18] S. Rosen, C. Harmonosky, and M. Traband, "Optimization of systems with multiple performance measures via simulation: Survey and recommendations," *Comput. Ind. Eng.*, vol. 54, no. 2, pp. 327–339, 2008.
[19] H. Beyer and B. Sendhoff, "Robust optimization-a comprehensive survey," *Comput. Method. Appl. M.*, vol. 196, no. 33-34, pp. 3190–3218, 2007.
[20] W. Abohamad and A. Arisha, "Optimisation methods in supply chain applications: A review," in *Proc. 12th annu. Irish Academy of Management Conf.*, Galway - Mayo Institute of Technology, Ireland, 2009.
[21] X. Franch and J. Carvallo, "Using quality models in software package selection," *IEEE software*, vol. 20, no. 1, pp. 34–41, 2003.

Waleed Abohamad, M.Sc. is a researcher in the 3S group (Research unit in Dublin Institute of Technology (DIT) specialized in complex systems simulation and optimization). He joined the 3S group in 2008 having spent four years as a senior researcher in Cairo University-Egypt where he received his B.Sc. and M.Sc. degrees in Computer Science. Currently, he is a PhD student at DIT. His research interests include Modeling and Simulation, Optimization, Computational Intelligence, and Machine Learning.

Amr Arisha, Ph.D. is a manager of the 3S Group (Research unit in DIT specialized in complex systems simulation and optimization) and a lecturer in School of Management in Dublin Institute of Technology. He joined DIT in 2005 having spent five years as a senior researcher in Dublin City University. Intel-Ireland has sponsored his research from 2000 - 2005. He received his Ph.D. in Industrial Engineering from Dublin City University (DCU). Dr. Arisha has published many journal and international conference articles in the area of manufacturing systems simulation and scheduling. His research interests include analysis and optimization of complex dynamic systems in manufacturing and service industries using stochastic simulation modeling, and optimization techniques. His research also includes Supply Chain Management, Operations Excellence, Project Management, Virtual Reality applications in Business Process Analysis, and Optimization for Process Improvement. He is a member in IIE, IMECH, IEI, ESE, ORS, IEEE and ASME.