International Journal of Computational Intelligence Systems, Vol. 7, No. 6 (December 2014), 1054-1063

A New Strategic Risk Reduction For Risk Management

Mourad Ykhlef
College of Computer and Information Sciences, Information System Department
King Saud University, Kingdom of Saudi Arabia
E-mail: ykhlef@ksu.edu.sa

Danah Algawiaz
Information System Department
Shaqra University, Kingdom of Saudi Arabia
E-mail: dalgawiaz@su.edu.sa

Received 9 June 2013

Accepted 4 March 2014

Abstract

Risk Management is one of the key cares of any organization strategic management; proper benefit of risk management is finding risks and their solutions. In this article, we will suggest a new Strategic Risk Reduction technique for producing optimal risk reduction strategies; which reduce risk exposure for expected income by allowing several countermeasures per risk rather than one countermeasure as previous works did. Our Strategic Risk Reduction will be optimized using Ant Colony Optimization approach.

Keywords: Strategic Risk Reduction, Strategic Method, Defect Detection and Prevention, Risk Reduction Leverage, Ant Colony Optimization.

1. Introduction

Risk management is an important step in project management because every project is a temporary endeavor undertaken to provide a unique result [1]; it is an undertaking that has not been done before. Therefore, all projects involve some level of risk, even if similar projects have been completed successfully; risk is a situation that, if it occurs, adversely affects the goals of a project [2]. Risk management is important for any project because of the uncertainties that all projects face. These uncertainties come from weakly defined requirements, changes in customer needs, difficulties in estimating the resources and the time needed for development and wide range in individual skills. A risk management process consists of many steps; risks identification is the first step, risk identification can begin with the source of problems, or with the problem itself. After risks have been identified, then the next step will be the assessment of the risk impact and its probability of occurrence. These can be difficult to know for sure in the case of the probability of an unwanted event occurring. Therefore, it is critical to make the excellent educated decisions in the assessment process in order to properly prioritize the implementation of the risk management plan. The most widely accepted formula for risk quantification is: Risk magnitude equal the impact of the event multiplied by probability of occurrence [3]. After identifying and assessing risks, the next step is risk planning process which considers each of the risks that have been identified, and develops strategies to manage these risks. For each risk, there is an action to take in order to minimize the disruption to the project if the trouble identified in the risk occurs. The methods of risk management typically include reducing the negative effect or probability of the risk, transferring the risk to another company, avoiding the risk, or even accepting some or all of the potential or actual effect of a particular risk [3]. Risk controlling is the stage immediately after risk plan phase. Risk controlling consists of preparing a Risk Treatment Plan, which documents the decisions about identified risks and how should be handled. The next step is implementation of the controlled methods for reducing the effect of the risks [4]. Risk management is a prioritization process that followed by how the risks with the highest loss (or impact) and the highest probability of occurring are carry first, and risks with not high probability of occurrence and lesser loss are handled in descending order [5]. “Risk reduction strategy” is the approach to make this selection [6]. The two techniques for obtaining effective risk reduction strategies are: the Defect Detection and Prevention (DDP) tool [7] and the Strategic Method [8]. We will propose a news risk reduction method that we name Strategic Risk Reduction (SRR); our new method takes advantages of both strategies [10]. Strategic method can't prevent or reduce risk in an efficient way because it reduces risk exposures by applying one countermeasure for each risk. So, it doesn't give the best strategy for a given budget. The Strategic Risk Reduction method, that we propose, overcomes weakens of the Strategic Method by allowing many countermeasures to be used for reducing a risk; unlike Strategic Method that reduced any risk by using only one countermeasure per risk. Defect Detection and Prevention (DDP) reduces risk exposures by applying sets of countermeasures for each risk. Also, it uses meta-heuristics to output best sets (may be optimal) of countermeasures. But it doesn't output the strategy which reduces risk exposures for a given budget. On the other hand, our Strategic Risk Reduction (SRR) reduces risk exposures by applying sets of countermeasure for each risk. Also, it provides best strategy (may be optimal) for a given budget. Strategic Risk Reduction overcomes drawback of Defect Detection and Prevention by providing a strategy as output rather it determines a set of attribute and countermeasure pairs that optimize risk reduction with respect to fixed budget. Our Strategic Risk Reduction is a method for producing optimal risk reduction strategies, that reducing risk exposure for expected income and allowing several countermeasures per risk. Finding optimal risk reduction strategy is a combinatorial process. To overcome the problem of exponential time of execution induced by Strategic Risk Reduction, we propose to optimize our Strategic Risk Reduction Algorithm with Ant Colony (SRR-ACO); SRR-ACO will use Ant colony Optimization to find best (may be optimum) subsets of countermeasures for each attribute (or risk) but in reasonable time of execution. The SRR-ACO utilizes the property of Evolutionary Algorithms that discovers new subsets in a polynomial time; it does not require knowledge about all subsets, this way our SRR-ACO can efficiently explore the space of possible subsets.

The reminder of this article is organized as follows: section 2 is devoted to explain Defect Detection and Prevention method. However section 3 will explain Strategic Method. Our Strategic Risk Reduction (SRR) method is detailed in section 4. Then section 5 will explain how to use Ant Colony Optimization for Strategic Risk Reduction (SRR-ACO). Experiments of our new method are presented in section 6. Finally we conclude in section 7.

2. Defect Detection And Prevention

The Defect Detection and Prevention (DDP) method is a powerful tool for choosing the various risks impacting requirements and the associated costs of relieving their influences [7]. The information gained through this method can be applied in many statuses where choices have to be adjusted in the earlier stages of a project. The application of DDP method could therefore be very useful in many hard or software production related projects. Detection and Prevention (DDP) have developed and implemented at Jet Propulsion laboratory, the DDP process used for achieving life-cycle risk management, and is a top-down approach to managing risk, this process has been represented in a software tool [9]. The Defect Detection and Prevention approach is applied as a way of making sure that selections made in the early stages of a project, are based on quantified information and probabilistic assessment methods, instead of incomplete human assessment. By using the Defect Detection and Prevention methodology, minimum costs to a project can be measured in early stages, as well as optimal risk distribution. In essence, the Defect Detection and Prevention method shares many characteristics with risk management systems, because risks are defined and assess in order to make a quantified and probabilistic assessment of their impact and likelihood. DDP then uses either exhaustive search when the number of techniques is small or heuristic search to locate near-optimal solutions (the current DDP uses simulated annealing [9]. Our Strategic Risk Reduction method will overcome weakens of the Defect Detection and Prevention by providing a strategy as output rather it determines a set of attribute and countermeasure pairs that optimize risk reduction with respect to fixed budget.

3. Strategic Method

The Strategic Method is a technique for producing optimal risk reduction strategies, which reduces risk exposure for expected income [8]. The input to the strategic method is a specification, about attributes (risks) and countermeasures which are risk reduction techniques of interest, the probability and impact of failure for each risk, both before and after using each countermeasure, and the cost of applying each countermeasure technique. The Strategic Risk Reduction method, that we propose, overcomes weakens of the Strategic Method by allowing many countermeasures to be used for reducing a risk; unlike Strategic Method that reduced any risk by using only one countermeasure per risk.

The strategic method uses a basic rule of risk exposure (*RE*), which is equal the product of the probability of loss *P(L)* and impact *S(L)*. Total system risk exposure is the sum of each risk exposures, *total RE* = $\Sigma P(L_i) * S(L_i)$, where L_i is the loss due to the i^{th} risk. The strategic method uses Risk Reduction Leverage (RRL). RRL is a simple measurement that gives a numeric value to a countermeasure, enabling different countermeasures to be compared. If Cost stands for the cost to implement countermeasure, then the formula for RRL is the change in Risk Exposure divided by the cost to implement countermeasure:

$$RRL = (RE_{before} - RE_{after}) / Cost$$

where RE_{before} is a risk exposure before using countermeasure, and RE_{after} is a risk exposure after using countermeasure.

$$RE_{before} = P_{before}(L) * S(L) \quad (1)$$

$$RE_{after} = P_{after}(L) * S(L) \quad (2)$$

It is usually the case that a countermeasure technique reduces only the likelihood of a risk and not its impact. In this case, the benefit is given by:

$$B = (RE_{before} - RE_{after}) - Cost$$
$$= [P_{before}(L) - P_{after}(L)] * S(L) - Cost \quad (3)$$

and the cost-benefit (CB) ratio reduced by *RRL* is given by:

$$CB = (RE_{before} - RE_{after})/Cost$$
$$CB = [P_{before}(L) - P_{after}(L)] * S(L)/Cost \quad (4)$$

The strategy generated is one that satisfies the utility function:

$$\min_{\tau,J}\left[\sum_{i=1}^{k} RE_{after}\left(A_i, T_{J(i)}\right) + CombCost(A_i, T_{J(i)}) + \sum_{i=k+1}^{N} RE_{before}(A_i)\right] \quad (5)$$

"where the minimum is taken over the sets *{(*A_1*,* $T_{J(1)}$*, τ(1)), (*A_2*,* $T_{J(2)}$*, τ(2)), …,(*A_N*,* $T_{J(N)}$*, τ(N))}* and all permutations *τ* of *{1,2,…,N}* and functions *J:{1,2,…,N}→{1,2,…,M}* (i.e. *J* is a set of non-distinct integers *1* through the number of activities *N*, *M* is the number of countermeasures. The utility function chooses the attribute-technique pairs that minimize total RE and cost after *k* activities have been performed assuming *k* is arbitrary" [8]. A risk profile (or RE profile) is the assessment of RE as a function of a monotonically increasing quantity such as elapsed time, cumulative effort, or cumulative cost [8].

For example [8] Table 1 and Table 2 (see Appendix for descriptions of attributes and techniques used here) show typical sets of risk and cost provided for calculating the effectiveness of countermeasures. Table 3 shows probability of a loss after assessing with a countermeasure. Table 4 shows the optimal strategy computed for the matrices in Tables 1, 2 and 3. In the first row of Table 4, risk reduction equal total RE because there is no countermeasures used to reduce risk, the risk reduction amount 15700 is resulted from 100∗6+90∗5+…+60∗40 (see Table 1). In second row upon the Strategic Method the first risk attached to the attribute A_{13} is reduced using countermeasure T_{11}. Risk exposures of A_{13} before using countermeasure T_{11} equal to 4500 (using formula 1) and risk exposures of A_{13} after using countermeasure T_{11} equal to 450 (using formula 2). Then risk exposure change of A_{13} using countermeasure T_{11} equal to 4500 – 450= 4050; the Cost of A_{13} using countermeasure T_{11} is equal to 10 (from Table 2) then benefit *B* of A_{13} using countermeasure T_{11} equal to 4040 (using formula 3) and cost benefit *CB* is equal to 405 (using formula 4). The risk reduction 11650 is the difference between the risk reduction of 15700 of the line 1 and the risk exposure change 4050.

The utility function result is equal to (3730+488+200=4418) (using formula 5) where 3730 is

RE_{after} for all assessed attributes; except attribute A12; using Tj countermeasures and 488 is cumulative cost for all assessed attributes; except A12; using Tj countermeasures, and 200 is the risk exposure of A12 (attribute not assessed). The attribute A12 is not assessed because all subsets of countermeasures will not reduce A12 probability or there is no benefit.

Table 1: Attributes And Their Loss Potential And Probability (Before Mitigation)

Attribute(i)	A1	A2	A3	A4	A5	A6	A7
loss potential for Ai	100	90	90	80	60	30	50
Pbefor(Ai)	6	5	20	15	20	5	20
Attribute(i)	A8	A9	A10	A11	A12	A13	A14
loss potential for Ai	20	10	10	60	10	90	60
Pbefor(Ai)	10	10	10	30	20	50	40

Table 2: Risk Assessment Techniques and the Costs of assessing them

Cost to assess A(i)w/T(j)	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
T1	50		10	70	10					50	5		10	
T2	100			100	100									
T3			80	80	80									
T4	100	90			19									
T5	70	100	70	70	70									
T6	30	30	30	30	30									
T7						5	10		5	5	3		3	
T8						80	70		80	80				
T9							3	10	20	20	20	10	20	10
T10	60			60	50	40	50	50	50	40	40	20	40	20
T11	60		90	60	60					50	10		10	
T12						5	5	10	10	10	10	5		
T13	30			30	30			30		30	5		30	
T14	100			100	100					100	5		100	

Table 3: Probability Of A Loss After Assessing With Technique T

Pafter	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
T1	4		15	12	15					5	15		20	
T2	6			13	15									
T3			15	12	13									
T4	6	0			19									
T5	6	2	2	13	18									
T6	6	2	5	13	19									
T7						2	15		8	10	30		30	
T8						1	10		7	9				
T9							10	4	6	8	25	20	30	30
T10	6			12	19	3	15	8	8	8	27	20	30	20
T11	3		15	5	5					5	5		5	
T12						3	18	9	10	10	30	20		
T13	5			12	15			5		6	20		28	
T14	3			3	5					5	10		20	

Table 4: The Optimal Strategy Computed For The Matrices In Tables (1- 3)

Attribute	Counter-mesure	RE before	RE after	RE Change	Cost	Benfit	CB	risk reduction	cumulati-ve cost
None	None	None	None	None	None	None	None	15700	0
A13	T11	4500	450	4050	10	4040	405	11650	10
A7	T9	1000	500	500	3	497	166.667	11150	13
A11	T11	1800	300	1500	10	1490	150	9650	23
A14	T10	2400	1200	1200	20	1180	60	8450	43
A3	T5	1800	180	1620	70	1550	23.143	6830	113
A6	T7	150	60	90	5	85	18	6740	118
A5	T11	1200	300	900	60	840	15	5840	178
A8	T9	200	80	120	10	110	12	5720	188
A4	T14	1200	240	960	100	860	9.6	4760	288
A2	T4	450	0	450	90	360	5	4310	378
A1	T11	600	300	300	60	240	5	4010	438
A9	T9	100	60	40	20	20	2	3970	458
A10	T13	100	60	40	30	10	1.333	3930	488

4. Strategic Risk Reduction

In this section we will present our new Strategic Risk Reduction (SRR) method that overcomes weakens of the Strategic Method and Defect Detection and Prevention at the same time. SRR allows many countermeasures to be used for reducing risk; unlike Strategic Method that is based on one countermeasure per risk. SRR overcomes drawback of Defect Detection and Prevention by providing a strategy as output rather it determines a set of attribute and countermeasure pairs that optimize risk reduction with respect to fixed budget. Our Strategic Risk Reduction (SRR) uses a basic rule of risk which is the risk exposure (RE) [11], RE is equal the product of the probability of loss *P(L)* and impact of loss *S(L)*. Total system risk exposure is the sum of each risk exposures, $total\ RE = \Sigma\ P(L_i) * S(L_i)$, where L_i is the loss due to the i^{th} risk. The SRR method uses Risk Reduction Leverage (RRL or Cost Benefit). Cost Benefit is a simple measurement that gives a numeric value to a countermeasure, enabling different countermeasures to be compared. It is usually the case that a technique reduces only the probability of loss *P(L)* of a risk and not its impact of loss *S(L)*.

In the following, for each attribute A_i; one can use a set of countermeasures $\check{T}_j$ such that $\check{T}_j = \{T_i \mid i \in \{1, \dots, n\}\}$, $1 \le j \le 2^n$. $CombCost\left(A_i, \check{T}_j\right)$ will stand for the combined cost to implement many countermeasures $\check{T}_j$ for an attribute A_i. The formula for Risk Reduction Leverage of any given set of countermeasures is the change in risk exposure divided by the combined cost to implement countermeasures:

$$RRL(A_i, \check{T}_j) = \Delta RE(A_i, \check{T}_j)/CombCost\left(A_i, \check{T}_j\right)$$

where risk exposure change is:

$$\Delta RE(A_i, \check{T}_j) = S(A_i) * P(A_i) * \left(1 - \prod_{T_k \in \check{T}_j} \frac{P(A_i, T_k)}{P(A_i)}\right)$$

where $P(A_i, T_k)$ is probability after assessment of A_i with T_k.

Given this information, we can perform the following algorithm to calculate an optimal risk reduction strategy:

Step 1: Label the greatest significant system assessment attributes from $A_1 \dots A_n$.

Step 2: Label the greatest significant system assessment countermeasures techniques sets from $\check{T}_1 \dots \check{T}_n$.

Step 3: Estimate the probabilities $P(A_i)$ and impact $S(A_i)$ with attributes A_i; $i=1 \dots n$ before any assessment

Step 4: Estimate the cost $C(A_i, T_k)$ and probability $P(A_i, T_k)$ after assessment of A_i with T_k, it is often the case that a countermeasures reduces only the likelihood of a risk and not its impact, the risk exposure after using the countermeasures techniques:

$$RE_{after}(A_i, \check{T}_j) = S(A_i) * P(A_i) * \prod_{T_k \in \check{T}_j} \frac{P(A_i, T_k)}{P(A_i)} \quad (6)$$

and the change in risk exposures:

$$\Delta RE(A_i, \check{T}_j) = S(A_i) * P(A_i) * \left(1 - \prod_{T_k \in \check{T}_j} \frac{P(A_i, T_k)}{P(A_i)}\right) \quad (7)$$

Step 5: Calculate the Combined cost:

$$\boldsymbol{CombCost\left(A_i, \check{T}_j\right) = \sum_{T_k \in \check{T}_j} Cost(A_i, T_k)} \quad (8)$$

and the benefit matrix:

$$B(A_i, \check{T}_j) = \Delta RE(A_i, \check{T}_j) - CombCost\left(A_i, \check{T}_j\right) \quad (9)$$

For each A_i find the $\check{T}j$ where $B(A_i, \check{T}_j)$ is maximum.

Step 6: Calculate the cost-benefit (CB) ratio reduced by RRL:

$$CB = \Delta RE(A_i, \check{T}_j)/CombCost\left(A_i, \check{T}_j\right) \quad (10)$$

sort *CB* from largest to smallest in the *CB* Column.

Step 7: Calculate Risk Reduction:

$$Risk\ Reduction = RE_{total} - \sum \Delta RE(A_i, \check{T}_j) \quad (11)$$

where:

$$RE_{total} = \Sigma P(L_i) * S(L_i) \quad (12)$$

Calculate cumulative cost for each attribute A_i located in row *n:*

$$CumulativeCost(1) = 0$$
$$CumulativeCost(n) = CumulativeCost(n-1) + CombCost\left(A_i, \check{T}_j\right) \quad (13)$$

The Strategic Risk Reduction strategy generated is one that satisfies the utility function:

$$\min_{\tau,J}\left[\sum_{i=1}^{k} RE_{after}\left(A_i, \check{T}_{J(i)}\right) + CombCost\left(A_i, \check{T}_{J(i)}\right) + \sum_{i=k+1}^{N} RE_{before}(A_i)\right] \quad (14)$$

where the minimum is taken over the sets $\{(A_1, \check{T}_{J(1)}, \tau(1)), (A_2, \check{T}_{J(2)}, \tau(2)), \dots, (A_N, \check{T}_{J(N)}, \tau(N))\}$ and all permutations τ of $\{1,2,\dots,N\}$ and functions $J:\{1,2,\dots,N\} \rightarrow \{1,2,\dots,M\}$ (i.e. J is a set of non-distinct integers 1 through the number of activities N, M is the number of countermeasures). The utility function chooses the attribute-technique pairs that minimizes total RE and combined cost after k activities have been performed assuming k is arbitrary.

Table 5 shows the optimal strategy computed for the matrices with two countermeasures techniques. For example the risk exposure RE_{after} of A13 after performing T7 and T11 is equal to (90 * 50 *(5/50)*(30/50) = 270) (using formula 6) where impact of A13 is S(A13)=90 and probability of A13 is P_{before} (A13)=50 (from Table 1), the probability of A13 after performing T7 is P_{after}(A13,T7)=30 and the probability of A13 after performing T11 P_{after}(A13,T11)=5 (from Table 3). The Risk exposure change of A13 after performing T7 and T11 equal to (90*50*(1-((5/50)*(30/50)) =4230) (using formula 7)). From Table 2, the cost of assessing A13 with T7 (resp. T11) is 3 (resp. 10); therefore the CombCost, after using T7 and T11 with A13 equal to 13 (using formula 8). Benefit for A13 after using T7 and T11 equal to (4230–13=4217) (using formula 9). CB ratio for A13 after using T7 and T11 equal to (4230/13=325.385) (using formula 10). RE_{total} is equal to 15700 (by applying formula 12 on Table 1, 15700 = 100∗6 + 90∗5 + … + 60∗40). Risk reduction equal to (15700–4230=11470) (using formula 11). The Cumulative Cost is 13 (using equation 13). The utility function result is equal to 4119 (3413+506+200) (using formula 14) where 3413 is RE_{after} for all assessed attributes using *Ťj* sets and 506 is cumulative cost for all assessed attributes (except *A12)* using Ťj sets, and 200 is the risk exposure of non assessed *A12*. The attribute A12 is not assessed because all subsets of countermeasures will not reduce A12 probability or there is no benefit. The utility function obtained by using our Strategic Risk Reduction method is 4119, which is larger better than the Strategic Method that has 4418 as optimum utility function result.

Table 5: The Optimal Strategy Computed For The Matrices In Tables (1 - 3).

Attribute	Countermesure	RE before	RE after	RE Change	Cost	Benefit	CB(RRL)	risk reduction	cumulative cost
None	None	None	None	None	None	None	None	15700	0
A13	T11 , T7	4500	270	4230	13	4217	325.385	11470	13
A11	T11	1800	300	1500	10	1490	150	9970	23
A14	T10	2400	1200	1200	20	1180	60	8770	43
A7	T9 , T7	1000	375	625	13	612	48.077	8145	56
A3	T5	1800	180	1620	70	1550	23.143	6525	126
A6	T7	150	60	90	5	85	18	6435	131
A5	T11	1200	300	900	60	840	15	5535	191
A8	T9	200	80	120	10	110	12	5415	201
A4	T14	1200	240	960	100	860	9.6	4455	301
A2	T4	450	0	450	90	360	5	4005	391
A1	T11	600	300	300	60	240	5	3705	451
A9	T9 , T7	100	48	52	25	27	2.08	3653	476
A10	T13	100	60	40	30	10	1.333	3613	506

5. Optimizing Strategic Risk Reduction Using Ant Colony Optimization

In the previous presentation we have seen that each risk may have from *1* to 2^n countermeasures subsets where *n* is number of countermeasures; this means that we are in the presence of an exponential problem. In this section we will present how to optimize our Strategic Risk Reduction using Ant Colony Optimization (SRR-ACO) that will be used in discovering the best (may be optimal) subsets of countermeasures for different risks in polynomial time.

Ant Colony Optimization (ACO) is a meta-heuristic, meaning that it's a general framework that can be used to create a specific algorithm to solve a specific exponential problem. Although ACO was proposed in a 1991 doctoral thesis by M. Dorigo, the first detailed description of the algorithm is given in [12]. ACO algorithm is an artificial intelligence technique based on the pheromone-laying behavior of ants; it can be used to find solutions to exceedingly complex problems that seek the optimal subset through a huge number of subsets. The SRR-ACO uses ants; each ant represents a potential subset, and each subset has to be associated with a particle pheromone including benefit of countermeasures subset to attribute (risk). SRR-ACO requires the specification of several parameters such as the pheromone influence factor α, the benefit influence factor ß, the pheromone evaporation coefficient ρ and proportion of ant probability Ǫ after using countermeasure subset Ťj; all these parameters control the behavior of the SRR-ACO algorithm.

The ants are initialized to random subsets; after initialization, the best ant has a maximum benefit to the attribute. The key idea of ACO is the use of simulated pheromones, which attract ants to better subset of countermeasures among the huge number of subsets. The main processing loop alternates between updating the ant subset based on the current pheromone values and updating the pheromones based on the new ant countermeasure subset. After the maximum number of iteration is reached through the main processing loop, the program displays the best countermeasure subsets found for each risk and its corresponding benefit using formula 9, risk exposure change using formula 7, cumulative cost using formula 13, and CB (risk reduction leverage) using formula 10. Finally SRR-ACO calculates utility function based on subsets that it chooses as best subset satisfying the formula 14.

Input:
N: number of attributes (risks);
T: maximum number of iterations;
NA: number of ants;
α=3, ß=2, ρ=0.01, and Q̄ =2 : Global parameters;

Output:
Best-subset(A_i) for each attribute Ai;
Best-utility;

SRR-ACO ( )
1. i=0; t=0;
2. **FOR** each attribute A_i, i<N
2.1. Initialize all Pheromone(A_i)=0.01;
2.2. **FOR** each t, t<T
2.2.1. k=1;
2.2.2. **FOR** each Ant(k), k<=NA
2.2.2.1. r = random number between 1 and 2^N;
2.2.2.2. $\check{T}_j$= binary representation of r;
2.2.2.3. Calculate $B(A_i, \check{T}_j)$; /* using formula 9 */
2.2.2.4. $\tau(A_i)$ = (Pheromone $(A_i))^\alpha * (B(A_i, \check{T}_j))^\beta$;
2.2.2.4.1. **IF** (k=1 or $\tau(A_i)$ > Best-$\tau(A_i)$) **THEN**
2.2.2.4.1.1. Increase (A_i) = Q̄ / $P(A_i, \check{T}_j)$;
2.2.2.4.1.2. Best−$B(A_i)$= $B(A_i, \check{T}_j)$;
2.2.2.4.1.3. Best−$\tau(A_i)$= $\tau(A_i)$;
2.2.2.4.1.4. Best-subset(A_i) = $\check{T}_j$
2.2.2.4.1.5. $CB(A_i) = CB(A_i, \check{T}_j)$; /* using formula 10*/
2.2.2.5. **END IF**
2.2.2.6. Decrease(A_i) = (1-ρ) * Pheromone (A_i);
2.2.2.7. Pheromone(A_i) = Increase (A_i) + Decrease(A_i);
2.2.3. k++;
2.2.4. **END FOR**
2.3. t++;
2.4. **END FOR**
3. i++;
4. **END FOR**
5. Sort attributes A_i according to $CB(A_i)$ from largest to smallest;
6. Compute best utility Best-utility using formula 14

Figure 1: SRR-ACO pseudo code.

The pseudo code of SRR-ACO is shown in Figure 1; SRR-ACO algorithm works as follows: the global parameters are first initialized by α=3, ß=2, ρ=0.01 and Q̄ =2. For each attribute A_i and for each ant among the set of underlying ants generate a random number. The binary representation of the random number is calculated to obtain a subset of countermeasures $\check{T}_j$; for example, for the random number 3 and its binary representation 00000000000011 the entailed subset of countermeasures is $\check{T}_j=\{T_1,T_2\}$. Based on pheromone and benefit values $\tau(A_i)$ associated with each $\check{T}_j$; apply pheromone update. Finally we sort the attributes A_i with their good subset $\check{T}_j$ from largest to smallest based on their cost benefit (risk reduction leverage) values. The underlying utility function is computed using formula 14.

As you can see from SRR-ACO, it took advantages of ACO methods. ACO, in common with other meta-heuristics, is quite sensitive to choice of free global parameters α, ß, ρ and Q̄. Although there has been quite a bit of research on ACO parameters, the general consensus is that you must experiment a bit with free parameters to get the best combination of performance and solution quality. In the next section, an experiment will be conducted to show how SRR-ACO doesn't take a long time to operate, to give best (may be optimal) set of countermeasures for each attribute A_i.

6. SRR-ACO Testing

SRR-ACO is applied to discover the best (may be optimal) subsets of countermeasures for different risks. SRR-ACO using basic rules of risk which are the risk exposure and risk reduction leverage (cost benefit). Table 1 and Table 2 show typical sets of risk and cost provided for calculating the effectiveness of techniques and Table 3 has the probability of a loss after assessing with a technique (countermeasure). All the experiments were performed on 2.53 GHz Intel® Core™ i5 CPU machine with 4.00GB RAM, running Microsoft Windows 7 Professional. The SRR-ACO algorithm is written with Java in NetBeans IDE 6.9.1 environment.

In the experiments, there are two parameters that have been determined: number of ants and maximum number of iteration. We conducted two experiments in order to inspect the speed and utility of SRR-ACO. First experiment set maximum number of repetitions to 150 with number of ants in [1…100]. However, the second experiment set the number of ants to 14 with maximum number of iteration in [15…1000]. The output of SRR-ACO experiments assures always the best subsets of countermeasures for each attribute or risk.

In first experiment the number of ants was set to 14 and maximum number of iterations is set to 150. Table 6 shows a sample of the best subsets generated from this experiment with utility equal to 3809.45.

Table 6 : Experiment#1 Output Sample

Attribute	Countermeasure	RE Change	Comb Cost	Benfit	CB(RRL)	risk reduction	cumulati-ve cost
None	None	None	None	None	None	15700	0
A14	T10	1200	20	1180	60	14500	20
A13	T1,T7,T11,T14	4456.8	123	4333.8	36.234	10043.2	143
A11	T1,T9,T10,T13,T14	1650	75	1575	22	8393.2	218
A6	T7	90	5	85	18	8303.2	223
A8	T9	120	10	110	12	8183.2	233
A2	T7	270	30	240	9	7913.2	263
A3	T1,T5,T6,T11	1774.6875	200	1574.688	8.8734	6138.5125	463
A7	T8,T9,T10	812.5	123	689.5	6.6057	5326.0125	586
A5	T1,T10,T11,T14	1146.5625	220	926.5625	5.21165	4179.45	806
A1	T11	300	60	240	5	3879.45	866
A4	T1,T11,T14	1136	230	906	4.94	2743.45	1096
A9	T9	40	20	20	2	2703.45	1116
A10	T13	40	30	10	1.33333	2663.45	1146

Figure 2 shows the result of this experiment related to the increase of number of ants. It shows also the time in seconds spent by SRR-ACO algorithm and the utility result of its final output. From figure 2, one can see that increasing the number of ants will increase execution time, because its termination condition depends on number of ants. Moreover, this increasing in the number of ants will most likely improve utility result of the final output.

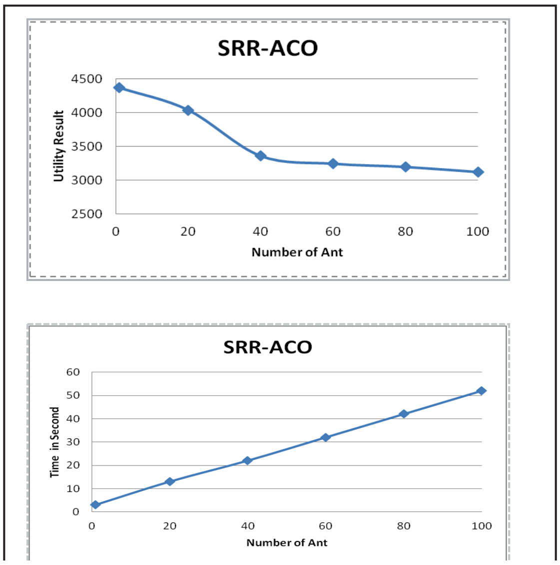

Figure 2: Time and utility related to number of ant

In the second experiment the number of ants was set to 14 and number of iterations was varied up to 1000.

Table 7: Experiment#2 Output Sample

Attribute	Countermeasure	RE Change	Comb Cost	Benfit	CB (RRL)	risk reduction	cumulative cost
None	None	None	None	None	None	15700	0
A14	T10	1200	20	1180	60	14500	20
A11	T1,T9,T11,T13,T14	1772.22	45	1727.22	39.383	12727.78	65
A13	T7,T9,T10,T11	4497.9	146	4351.9	30.82	8229.88	211
A3	T1,T5	1789.875	160	1629.875	11.187	6440.005	371
A5	T1,T11,T13	1031.25	100	931.25	10.3125	5408.755	471
A6	T7,T12	141.36	20	121.36	7.068	5267.395	491
A7	T7,T8,T9,T12	971.5234	176	1795.523	5.52	4295.8716	667
A4	T5,T11,T14	1130.67	230	900.67	4.9159	3165.2016	897
A8	T9,T13	160	40	120	4	3005.2016	937
A2	T5,T7	378	120	258	3.15	2627.2016	1057
A1	T1,T13,T14	433.3	180	253.3	2.407	2193.9016	1237
A9	T9	40	20	20	2	2153.9016	1257
A10	T13	40	30	10	1.33333	2113.9016	1287

Table 7 shows a sample of the best subsets generated from this experiment with utility equal to 3400.

Figure 3, shows the result of this experiment related to the number of iteration. It shows the time, in seconds spent by the SRR-ACO algorithm and the utility result of its final output. From figure 3, one can see that increasing the number of iterations will most likely increase the execution time; because it takes more time to apply the ACO cycles and calculate the subset variables of each ant. Moreover, this increasing in the number of iteration guarantee improvement of the utility result and it could even make it better.

In order to suggest the best values of number of ants, and maximum number of iteration for SRR-ACO; we interrelate the two experiments. Considering the time in the two experiments, in figure 2, and figure 3, you can see that increasing number of time for iterations slow down ACO more than increasing the number of ants. Considering the utility results in the two experiments given in figure 2 and figure 3, you can see that increasing number of ants and number of iteration will improve utility results. Considering the output samples of the two experiments, in Table 6, and Table 7, you can see that the best utility result of the second experiment (3400) is better than the best utility result of the first experiment (3809.45).

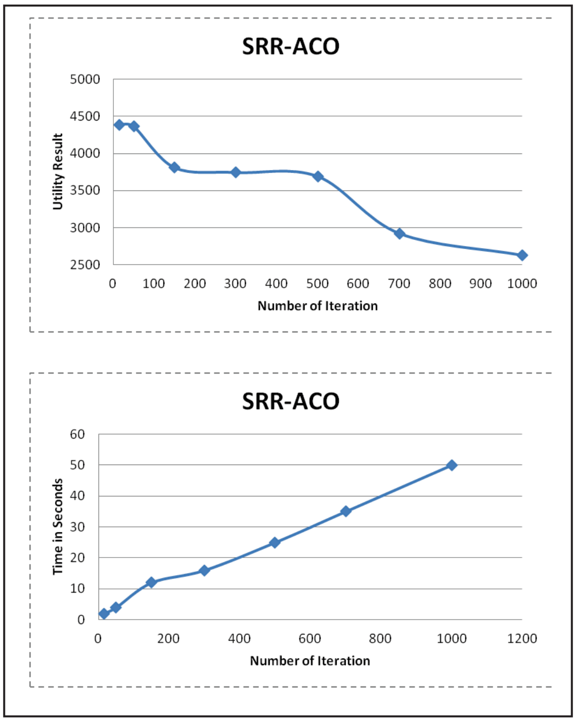

Figure 3: Time and utility related to number of iteration

From all the experiments, it is observed that increasing number of ants will take less time than increasing number of iterations but it will hurt utility result. In addition, after comparing the best output results in figure 2, and figure 3, it is observed that using of high number of iteration will generate almost better utility results but it will increase execution time, i.e. high iteration will acquire more time. At the end, from the results given above, it can be seen that it is important to choose the right numbers of ants, and maximum number of iterations. According to our experiments with SRR-ACO, it is better to use high iteration (e.g. 150) with moderate number of ants (e.g. 14). This tradeoff decision will prospectively guarantee good results in short time.

7. Conclusion

In this article, we introduced a new Strategic Risk Reduction Algorithm that overcomes weakens of the Strategic Method and Defect Detection and Prevention. Our SRR method allows many countermeasures to be used for reducing risk impact; unlike Strategic Method that reduces risk impact by considering only one countermeasure for each risk. Strategic Risk Reduction overcomes also the drawback of Defect Detection and Prevention by providing a strategy as output rather it determines a set of attribute and countermeasure pairs that optimize risk reduction with respect to fixed budget. Our Strategic Risk Reduction (SRR) method was also augmented with Ant Colony Optimization Algorithm (SRR-ACO) to find best subsets of countermeasures for each risk or attribute in polynomial time. SRR-ACO experimental results demonstrated that SRR-ACO algorithm, incorporate high performance concerning the time of execution.

Acknowledgement

This work was supported by Deanship of Scientific Research and Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

References

[1] "Project Management Institute," in *A Guide to the Project Management Body of Knowledge (PMBOK Guide),Fourth Edition*, (ANSI/PMI 99-001-2008), 2008, pp. 273-312.

[2] P. R. Garvey, Analytical Methods for Risk Management: A Systems Engineering Perspective, LONDON : Chapman-Hall/CRC-Press,Taylor & Francis Group (UK), October 20, 2008 .

[3] M. S. Dorfman, Introduction to Risk Management and Insurance (9 ed.), Englewood Cliffs, N.J: Prentice Hall. ISBN 0-13-224227-3, 2007.

[4] N. Crockford, in *An Introduction to Risk Management (2 ed.)*, Cambridge, UK: Woodhead-Faulkner,ISBN 0859413322, 1986, p. 18.

[5] I. Sommerville, Software Engineering (9th Edition), Addison Wesley; 9 edition ,ISBN-10: 0137035152, March 13, 2010.

[6] J. D, C. Richardson, D. Port and M. S, "Exploring the Robustness of Risk Reduction Strategies," in *17th IEEE International Symposium on Software Reliability Engineering*, North Carolina, 2006.

[7] S. Conford, "Managing Risk as a Resource using the Defect Detection and Prevention process," *IEEE SOFTWARE Probibalistic Safety Assessment and Management,* pp. 1609-1614, 1998.

[8] D. P. a. M. F. J. Richardson, "Robust Derivation of Risk Reduction Strategies," in *IEEE*, Montana, 2007.

[9] M. Ykhelf and D. ALGawiaz, "A New Strategic Risk Reduction," in *5th International Conference on Computer Science and Information Technology*, Amman,Jordan, 2013.

[10] M. S. F. &. K. A. H. Steven L. Cornford, "DPP: A Tool for Life-Cycle Risk Management," *IEEE A&eSYSTEM Magazine,* 2006.

[11] K. Schwalbe, Information Technology Project Management, ISBN-10:0-324-78692-1, 2009.

[12] M. D. &. V. M. Colorni A., "Distributed Optimization by Ant Colonies," in *Proceedings of the First European Conference on Artificial Life*, Proceedings of the First European Conference on Artificial Life, 1992.

Appendix A. Appendices

The attribute and technique descriptions for the data indicated in the tables are [8]:

A1:	Robustness/Independent redundancy (No Single Failure Point, Priority Inversion)
A2:	Robustness/Independent redundancy (No Single Failure Point, Requirement Consistency, Completeness)
A3:	Robustness/Independent redundancy (No Single Failure Point, Code Quality)
A4:	Stability of Performance (Timing, Message queue over flow)
A5:	Real time performance (Don't skip the data flame)
A6:	Development Schedule
A7:	Cost
A8:	Portability/ Replace-ability, Adaptability (to Hardware or Driver)
A9:	Maintainability/Changeability
A10:	Scalability (Capability of adding application code)
A11:	Testability
A12:	Understandability (access to code)
A13:	Resource Utilization (How much resource used when maximum process is on using past system
A14:	Vender Support (Response time)
T1:	Test suites
T2:	Analysis Using Model
T3:	API Test
T4:	Model Checking
T5:	Code Review Lessons Learned
T6:	Static Analysis of code
T7:	Estimation
T8:	Custom Method
T9:	Interview Vendor
T10:	Investigation of past data
T11:	Test on Emulator
T12:	Best Guess
T13:	Benchmark test
T14:	Simulation