Global Journal of Computer Science and Technology
Volume 12 Issue 3 Version 1.0 February 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Iteration Based Risk Tracker Evolutionary Algorithm with Component Based Development

By Dr.S.Murali Krishna &Saroj Kumar Gupta
Madanapalle Institute of Technology and Science

Abstract - This paper presents a component based development (CBD) model for each iteration of software development. Risk management is a major part of software planning and risk tracking is one of the important functions of risk management and without proper tracking, it is quite difficult to control risk. Software industry is more deeply in recent years about it. From the beginning of the development to the end of the development assessing the software quality and assurance of software quality become an important, with this regards in this proposed paper a novel evolutionary algorithm is used for a feasible solution.

Keywords : TELOS, Risk Analysis, Risk Tracker, CBD.

GJCST Classification: D.2.2

ITERATION BASED RISK TRACKER EVOLUTIONARY ALGORITHM WITH COMPONENT BASED DEVELOPMENT

Strictly as per the compliance and regulations of:

Iteration Based Risk Tracker Evolutionary Algorithm with Component Based Development

Dr.S.Murali Krishna [α], Saroj Kumar Gupta [σ]

Abstract - This paper presents a component based development (CBD) model for each iteration of software development. Risk management is a major part of software planning and risk tracking is one of the important functions of risk management and without proper tracking, it is quite difficult to control risk. Software industry is more deeply in recent years about it. From the beginning of the development to the end of the development assessing the software quality and assurance of software quality become an important, with this regards in this proposed paper a novel evolutionary algorithm is used for a feasible solution.

IndexTerms : TELOS, Risk Analysis, Risk Tracker, CBD.

I. Introduction

In the last two decades, constrained optimization problems (COPs) have received much attention by most researchers and practitioners. The models for most real-world applications are established in the form of constraints imposed upon the objective function. In General, most of constraint handling techniques previously discussed will inevitably solve two important problems: (1) how to generate the feasible solutions, (2) how to direct the search to find the optimal feasible solution. And Risk management is also a major part of software planning and risk tracking is one of the important functions of risk management. Without proper risk tracking, it is quite difficult to control risk while developing any software. A component based model is presented for recognizing the characteristics of diverse information extracted from solutions. Simultaneously in the component based model, the interrelationship between each dimensional component (i.e. decision variable) of solution and constraints is revealed. A novel measurement of feasibility is defined[1]. Different from traditional measurement, the definition in this research paper is only related to a component of solution along with risk occurred during development phase and risk tracking is one of the proposed solution in this paper. The feasibility of components is measured so as to direct which component needs to be transformed at a lower cost.

Author α : Professor in CSE, Madanapalle Institute of Technology and Science, P.B.No.14, Angallu, Madanapalle - 517325, Chittoor District, Andhra Pradesh,India Email Id : drmuralimits@gmail.com

Author σ : Assistant Professor in CSE, Madanapalle Institute of Technology and Science, P.B.No.14, Angallu, Madanapalle - 517325, Chittoor District, Andhra Pradesh,India

Email Id : sarojkumargupta@gmail.com

II. Feasibility Factor (TELOS)

A feasibility study is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation. In other words, a feasibility study is an evaluation or analysis of the potential impact of a proposed project. Five common factors (TELOS)

a) Technology and System Feasibility

The assessment is based on an outline design of the system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological Five common factors (TELOS)

b) Technology and System Feasibility

The assessment is based on an outline design of the system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project.

c) Economic Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. Cost Based Study: It is important to identify cost and benefit factors, which can be categorized as follows: (1) Development costs; and (2) Operating costs. This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system. Time Based Study: This is an analysis of the time required to achieve a return on investments and the benefits derive from the system. The future value of a project is also a factor.

d) Legal Feasibility

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

e) Operational Feasibility

Is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

f) Schedule Feasibility

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interaction in between. The iterative and incremental development is an essential part of the Rational Unified Process, the Dynamic Systems Development Method, Extreme Programming and generally the agile software development frameworks[2][3].

III. Iterative Development

Iterative development slices the deliverable business value (system functionality) into iterations. In each iteration a slice of functionality is delivered through cross-discipline work, starting from the model requirements through to the testing deployment.

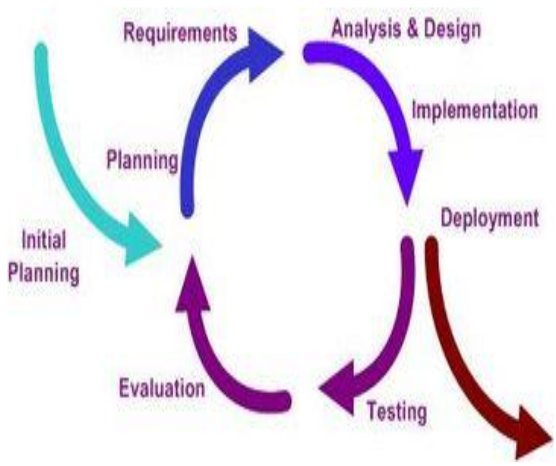

Fig. 1. Iterative cyclic software development process

The unified process groups Iterations into phases: Inception, Elaboration, Construction, and Transition.

- Inception identifies project scope, risks, and requirements (functional and non-functional) at a high level but in enough detail that work can be estimated.
- Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
- Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
- Transition delivers the system into the production operating environment.

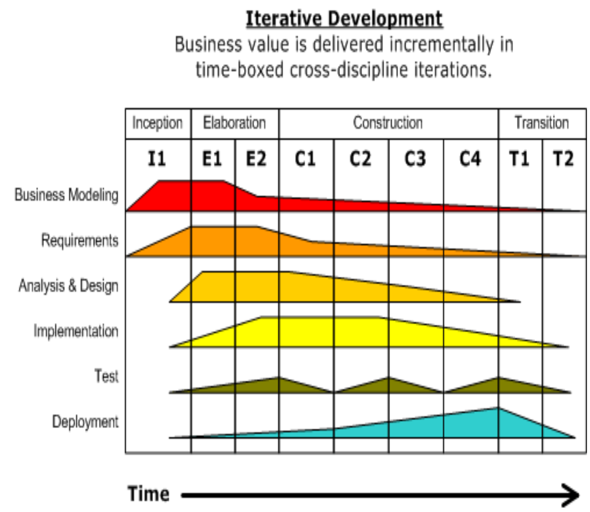

Fig. 2. Iterative software development

IV. Risk Analysis and Tracker

In development process risk analysis and management is a major part of software planning and risk tracking is one of the important functions of risk management without proper tracking, it is quite difficult to control risk. In this proposed model software risk tracking is an important function of software risk management. It is not sufficient to identify risk, prioritize them, generate risk plan and provide assessment about probability of risk but it is important to track them for controlling. So a strategy is proposed to development using risk tracking Pareto Distribution[4]. Risk tracking diagram is given below.

In development process risk analysis and management is a major part of software planning and risk tracking is one of the important functions of risk management without proper tracking, it is quite difficult to control risk. In this proposed model software risk tracking is an important function of software risk management. It is not sufficient to identify risk, prioritize them, generate risk plan and provide assessment about probability of risk but it is important to track them for controlling. So a strategy is proposed to development using risk tracking Pareto Distribution[4]. Risk tracking diagram is given below.

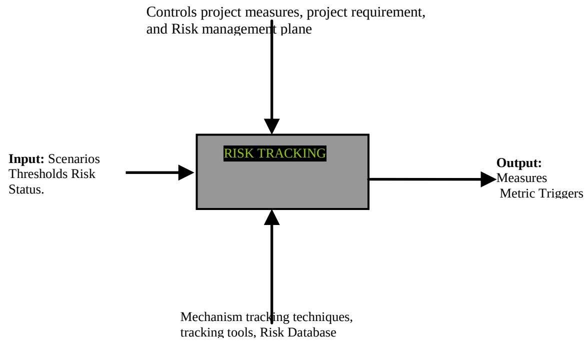

Fig 3. Risk Tracking

During data collection about the risk, top nine risks which effect the development of software are identified.

1. Business Domain,
2. Communication,
3. Customer,
4. Environmental/Natural cause,
5. Project Management,
6. People/ HR,
7. Quality Process,
8. Technology,
9. Infrastructure.

Algorithm steps for development of software risk tracker *(Risk tracker Algorithm)*

1. Input the data related to different risk type and categorized them into different classes.
2. Check the severity of risk based on risk description, it may be low, medium, high or very high.
3. Determine impact and Loss Expected (LE). It is calculated in rupees in terms of impact and their consequences. This may be one time of low impact, two times of medium, three times of high and four times of very high risk. LE = impact* 10K consequences Rs.
4. Now rank risk from 1 to 5 on the basis of highest to lower LE using Pareto Distribution. This will ranks the risk on the basis of LE.
5. Plot the bar chart of risk and their LE which considers impact and consequences.

V. The Proposed Algorithm (CMR)

A component based model is presented. Two main technologies are studied in this model extraction and partition technology. The former is used to realize which component of a solution needs to be transformed at a smaller cost, and the latter is for directing the transformation of the infeasible component into the feasible one.

a) Extraction Technology

For a constrained optimization problem, each constraint is only correlative with certain components, namely, each component only impacts upon some specific constraints. Therefore, the interrelationship between each dimensional component of solution and constraints is revealed.

$$g01:\ Min\ F(x) = 5\sum_{i=1}^{4} X_i - 5\sum_{i=1}^{4} X_i^2 - \sum_{i=5}^{13} X_i$$

$$g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0$$
$$g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0$$
$$g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0$$
$$g_4(x) = -8x_1 + x_{10} \le 0$$
$$g_5(x) = -8x_2 + x_{11} \le 0$$
$$g_6(x) = -8x_3 + x_{12} \le 0$$
$$g_7(x) = -2x_4 - x_5 + x_{10} \le 0$$
$$g_8(x) = -2x_6 - x_7 + x_{11} \le 0$$
$$g_9(x) = -2x_8 - x_9 + x_{12} \le 0$$

Where $0 \le x_i \le 1 (i = 1, ..., 9)$, $0 \le x_i \le 100 (i = 10, 11, 12)$ and $0 \le x_{13} \le 1$.

The optimum satisfied where $f(x^*) = -15$. The first dimensional component x_1 appears in some inequality constraints including g_1, g_2 and g_4. It means that the change to x_1 will impact on whether the constraints (g_1, g_2, g_4) are satisfied or not. By analogy, the correlative constraints of each dimensional component in this test function can be deduced in Table I. Supposing the correlative constraint set of the *ith* dimensional component is marked as CRG_i. The population size is represented as N_p and the dimension of constrained problem is D. After the above deduction, it is known that the *ith* dimensional component needn't to change when all correlative constraints are satisfied, or this component should change when there is at least one correlative constraint isn't satisfied. However, if the correlative constraint set is empty, it means that the feasibility measure of solution isn't relevant to this component. In order to determine whether the component to change, some definitions are given.

Definition 1 (Feasible Component)

For $x_i \in X$, $i = 1, ..., D$, $\forall g_j \in CRG_i$, if $max[0, g_j] = 0$,
then x_i is a feasible component.

Definition 2 (Infeasible Component)

For $x_i \in X$, $i = 1, ..., D$, $\forall g_j \in CRG_i$, if $max[0, g_j] \neq 0$,
then x_i is an infeasible component.

Definition 3 (Feasible Solution)

For X_i, $i = 1, ..., N_p$, $\forall x_j \in X_i$, if x_j is a feasible component, then X_i is a feasible solution.

Definition 4 (Infeasible Solution)

For X_i, $i = 1, ..., N_p$, $\exists x_j \in X_i$, if x_j is an infeasible component, then X_i is an infeasible solution.

According to these definitions, a feasible solution is constituted by feasible components, and an infeasible solution maybe constituted by feasible and infeasible components. Therefore, the infeasible solution merely needs to be modified its infeasible components for transforming into feasible solution; the rest feasible components don't need change.

b) Partition Mechanism

Based on the various dimensional components, partition mechanism is applied to direct the transformation of infeasible component into feasible one. The number of partition regions depends on the dimensions of constrained problem. Each dimensional partition region will conserve many solutions which contain the relevant dimensional feasible components. In other words, if the *ith* dimensional component of a solution is feasible, the *ith* partition region includes this solution. If the solution includes many various dimensional feasible components, it will be conserved in many related partition regions. The example of the above proposed test function is displayed in figure 4.

According to the number of solutions in each partition region and feasibility proportion of the current population, various kinds of multi-parent crossover operators are designed for solution feasibility or population diversity. An opposition based mutation with a probability is embedded in CMR to accelerate its convergence speed. The generation for the new offspring is shown in Algorithm 1. Let us define the meeting of the following terms: $N\,umf\,c_j$ is the number of solutions in the *jth* partition region; P_m is mutation rate and $P_f\,r$ is feasibility proportion of the current Population, $[M\,in^t_j, M\,ax^t_j]$ is the range of the *jth* component in the i*th* generation; rd_1, rd_2, rd_3 are different random numbers sampled in [-0.5,1.5], simultaneously, $rd_1 + rd_2 + rd_3 = 1$.

Algorithm1 Offspring Generation with Genetic Operators

for i = 0 to N_p **do**

for j = 0 to D **do**

if rand(0, 1) < P_{fr} $\wedge$ Num $fc_j \geq 3$ **then**

select three parents P_{r1j} , P_{r2j} , P_{r3j} randomly from the jth partition region;

$^{x}{}_{ij} = {}^{rd1P}r_1j + {}^{rd2P}r_2j + {}^{d3P}r_3j;$

else

select three parents P_{r1j} , P_{r2j} , P_{r3j} randomly from the current population;

$^{x}{}_{ij} = {}^{rd}1^{P}r_1j + {}^{rd2P}r_2j^{d3P}r_3j;$

end if

if rand(0, 1) $\leq P_m$ then

$x_{ij} = M\,in^t_j + M\,ax^t_j - x_{ij}$

end if

end for end for

c) The Novel Ranking Method

Ranking method presents a new view on balancing the dominance of penalty and objective functions. Many researchers have done some study on ranking methods with various lexicographic orders, such as stochastic ranking, Pareto ranking. In general, feasible solutions are ranked highest and better than all infeasible solutions. However, infeasible solutions with superior objective function value are more efficient to guide the population toward the optimum feasible, especially when the feasible regions are disjoint or the optimum lies on the boundary of the feasible region. Therefore, we tend to remain the important feasible and infeasible solutions. A novel ranking strategy is designed to accomplish the above goal. The essential comparison rules between adjacent pairs can be summarized as the following three points: 1) two feasible solutions are compared only based on their objective function values; 2) two infeasible solutions are compared only based on their objective function values, while at least there are one's objective function value less than the value of best feasible solution in the current population; 3) In the remaining situations, two solutions are compared based on the amount of their constraint violations[9]. After the comparisons, infeasible solutions with superior objective function value are ranked highest, followed by all feasible solutions and other infeasible solutions with greater constraint violation value are ranked to the lowest level.

Considering little feasible[5] solutions for the population at the early evolutionary stage, the ranking strategy should pay more attention to feasibility or constraint violation for a solution. So the whole ranking method is described in Algorithm 2. Where *Pr* is a proportion constant in [0, 1].

Algorithm 2. Ranking Method

if $P_{fr} \leq P_r$ **then**

compare the adjacent pair according to the amount of their constraint violations, regardless of feasible or infeasible solutions;

else

compare the adjacent pair according to the above rules 1),2),3);

end if

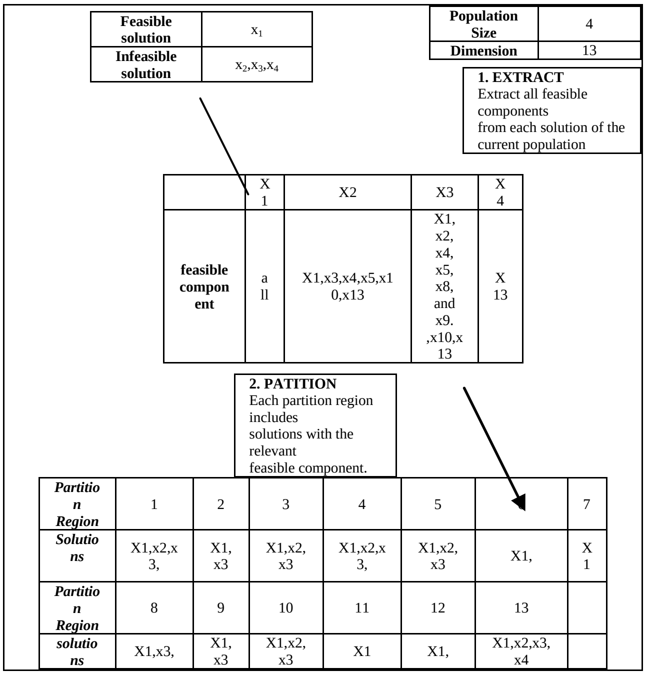

Fig 4. An example of partition Technology

VI. Experiment Verification

Four benchmark test functions are applied in this paper, and the results of the CMR algorithm are compared against three state-of-the-art algorithms: the SR(Stochastic Ranking), the KM (Koziel & Michalewicz), and the SAFF(Self Adaptive Fitness Formulation). For each test case, 30 independent runs are performed. In the following experiments, the parameters for the CMR algorithm are as follows: the population size $N_p = 60$, the maximum generations is 5000, the mutation rate $P_m = 0.25$ and $P_r = 0.3$, $\varepsilon = 10^{-4}$. The experiments are performed on a computer with Intel Core-2 CPU 1.83GHz and 1GB of RAM, by using the visual C++ compiler.

The rest benchmark functions $g02 - g04$ are described as following:

Table I

Statistical Results for G01- G04 Functions

Alg.	Best	Mean	Worst	st. dev
		G01(-15.0000)		
KM	-14.786	-14.708	-	-
SAFF	-15.000	-15.000	-15.000	0.0E+00
SR	-15.000	-15.000	-15.000	0.0E+00
CMR	-15.000	-15.000	-15.000	0.0E+00

Alg.	Best	Mean	Worst	st. dev
		G01(-15.0000)		
KM	-14.786	-14.708	-	-
SAFF	-15.000	-15.000	-15.000	0.0E+00
SR	-15.000	-15.000	-15.000	0.0E+00
CMR	-15.000	-15.000	-15.000	0.0E+00

In this section, the main steps of CMR algorithm can be described in figure 5.

$$g02 \ : M\inf(X) = -(\sqrt{2})^n \prod_{i=1}^{n} x_i$$

$$h(X) = \sum_{i=1}^{n} x_i^2 - 1 = 0$$

Where $n = 10$ and $0 \le xi \le 1$ ($i = 1, \ldots, n$). The optimum satisfied where $f(x^*) = -1$.

$$g03 : Min\ f(X) = -[sin3(2\pi x1)\, sin(2\pi x2)]/[x_1^3 (x1 + x2)]$$

$$\text{s.t. } g1(X) = x_1^2 - x2 + 1 \le 0$$

$$g2(X) = 1 - x1 + (x2 - 4)^2 \le 0$$

Where $0 \le x1 \le 10$ and $0 \le x2 \le 10$. The optimum satisfied

Where $f(x^*) = 0.095825$.

$$g04 : Min\ f(X) = x^2_1 + (x2 - 1)^2$$

$$h(X) = x2 - x_1^2 = 0$$

Where $-1 \le x1 \le 1$ and $-1 \le x2 \le 1$. The optimum satisfied

Where f $(x^*) = 0.75$.

Table 1 summarizes the results from the conducted experiment. The statistical results include the known optimal solutions for each test function, the *best, mean, worst* objective function values, and the standard deviations. "—" means that solutions were not found or not available. In the comparison, CMR can consistently find the optimal solutions in four test functions (g01, g02, g03, and g04) as other compared algorithms. All *best*, *mean* and *worst* objective function values of CMR were equivalent to the optimums for the above functions. Especially, CMR has better capability to deal with function g02 and has slightly better standard deviations than SR, SMES and others. The experimental results illustrate the performance of CMR algorithm is similar to the compared algorithms in terms of the solutions quality. With slightly better standard deviations, CMR is more robust and stable in obtaining consistent results than all the compared optimization algorithms. In all experiments, feasible solutions were continuously found for all the test functions in 30 runs. These results revealed that CMR has the substantial capability to deal with different kinds of *COPs*.

VII. Conclusion

In this paper we have presented a Iteration based risk tracker evolutionary algorithm with component based development, which is based on a component based model and a new ranking method. Extraction and partition studied in this model are two main technologies. The performance of this algorithm has been extensively investigated by experimental studies with the risk tracker applier on each iteration. The experimental results illustrate the CMR performance in terms of the quality of the resulting solutions, especially for robustness stability and feasibility in obtaining consistent results. In our future work , multi linear cost optimization process is studied for Extraction and Partition of various models.

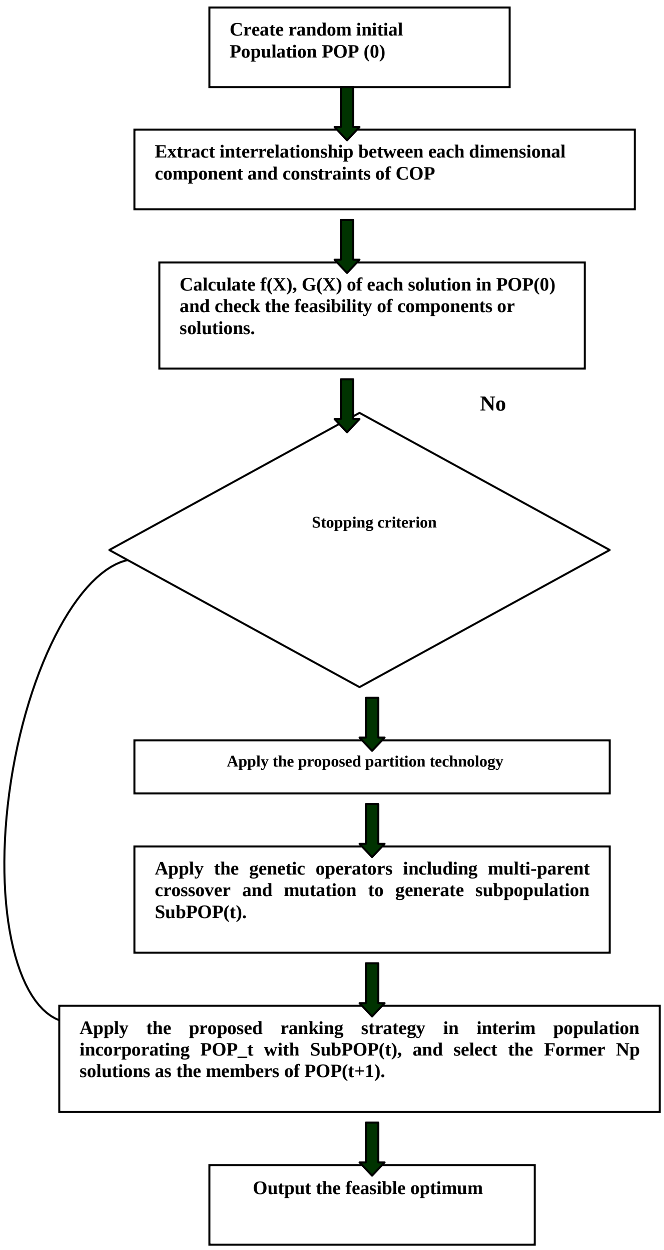

Fig. 5 The Flow Chart of CMR algorithm

References Références Referencias

1. William E. Levis, Software Testing and Continuous Quality improvement third edition, CRC Press.
2. Somerville, Software Engineering, seventh Edition, Pearson Education.
3. Z. Michalewicz and M. Schoenauer, *Evolutionary algorithm for constrained parameter optimization problems*, Evolutionary Computation
4. C. A. C. Coello, *Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art*, Computer Methods in Applied Mech.
5. S. Venkatraman and G. G. Yen,A generic framework for constrained optimization using genetic algorithms, IEEE Trans. Evolutionary Computation.
6. T. P. Runarsson and X. Yao, Search biases in constrained evolutionary optimization, IEEE Trans. Evolutionary Computation.
7. S. Rahnamayan and H. R. Tizhoosh and M. M. A. Salama, Oppositionbased differential evolution algorithms.
8. Shen LI , Yuanriang LI , Bo Wei and Yu WU , A Novel Evolutionary Algorithm with Component Based Model , IEEE …..
9. T. P . R unarson and X . Yao , Stochastic ranking for constrained evolutionary optimization , IEEE Trans. Evolutionary computation , vol . 4 , no . 3 , pp . 284 – 294 , sep 2000.