Redundancy Allocation for Series-Parallel Systems Using Integer Linear Programming

Alain Billionnet

ENSIIE, 18 allée Jean Rostand, F-91025, Evry cedex

{billionnet@ensiie.fr}

Abstract. We consider the problem of maximizing the reliability of a series-parallel system given cost and weight constraints on the system. The number of components in each subsystem and the choice of components are the decision variables. In this paper, we propose an integer linear programming approach that gives an approximate feasible solution, close to the optimal solution, together with an upper bound on the optimal reliability. We show that integer linear programming is an interesting approach for solving this reliability problem: the mathematical programming model is relatively simple; its implementation is immediate by using a mathematical programming language and an integer linear programming software, and the computational experiments show that the performance of this approach is excellent based on comparison with previous results.

Keywords: system reliability; redundancy allocation; multiple component choices; integer linear programming; computational experiments

1. Introduction

A system is a collection of components arranged to a specific design in order to achieve desired functions with acceptable performance and reliability. The types of components, their quantities, their qualities and the manner in which they are arranged within the system have a direct effect on the system's reliability. In a series configuration, a failure of any component results in failure for the entire system. In a parallel configuration at least one of the units must succeed for the system to succeed. Units in parallel are also referred to as *redundant* units. Redundancy is an important aspect of system design and reliability in that adding redundancy is one of several methods of improving system reliability. Another method consists in improving component reliability. Note that both methods result in an increase in system cost. While many systems can be represented by either a simple series or parallel configuration, other systems involve both series and parallel configurations.

Here we consider the redundancy allocation problem (RAP) in a system which is composed of a specific number of subsystems in series. Figure 1 presents an example of such a system. For each subsystem several functionally equivalent components are available. Each subsystem includes one or several components arranged in parallel. RAP consists in determining the number of components of each type which must compose the subsystem so that the reliability of the complete system is maximized, considering certain constraints such as the cost and the weight of the system.

The RAP is a well-known system reliability optimization problem which can be formulated as a difficult nonlinear integer program. It has been extensively studied and solved using many different mathematical programming and heuristic approaches.

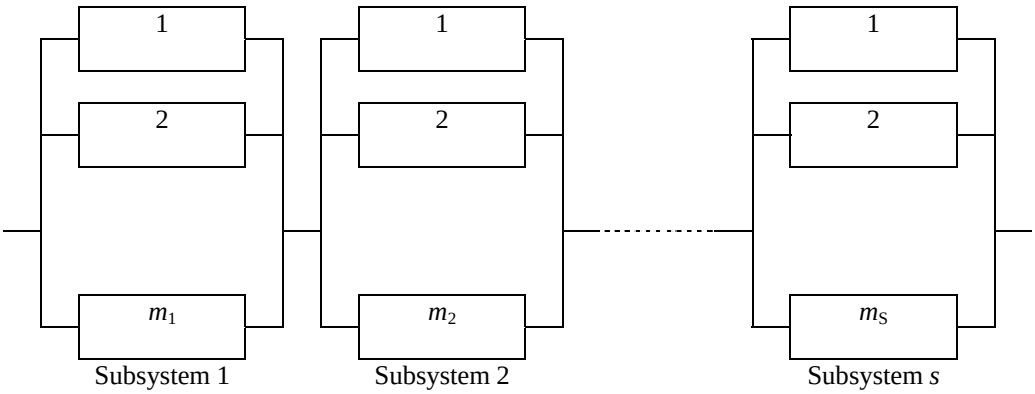

Figure 1. Example of series-parallel system

In this paper we show that integer linear programming is an interesting alternative for solving RAP. The integer linear programming formulation is easy to understand and to implement if one has a mathematical programming language together with an integer linear programming software. In Section 2 we formulate the problem and give notations. In Section 3 we review some works on RAP. In Section 4 we describe the integer linear programming approach we propose for RAP. Section 4 is devoted to computational experiments and Section 5 presents conclusions.

2. Formulation and notations

The problem objective is to maximize system reliability, $R(x)$, given constraints on the system cost and weight. The system is configured as a series-parallel system. The number of components in each parallel configuration, i.e. subsystem, and the choice of components are the decision variables. For each subsystem, there are m_i functionally equivalent components types available, each with different reliability, cost and weight. In this work, the components will be assumed to be statistically independent. That implies that failure of one component does not affect other components in the system.

Notations

s : number of subsystems in the system;

m_i : number of available component choices for subsystem i;

r_{ij}, c_{ij}, w_{ij} : reliability, cost and weight of component j available for subsystem i, respectively;

x_{ij} : number of component type j used in subsystem i (decision variables);

$C_{\max}$, $W_{\max}$: system cost, weight constraints limits;

$R(x)$, $C(x)$, $W(x)$: reliability, cost and weight of the system, respectively;

$R_i(x)$: reliability of subsystem i;

N : set of natural numbers.

It is well-known that RAP can be formulated as the following nonlinear integer program P1:

$$(P1)\begin{cases} \max \ \prod_{i=1}^{s}[1-\prod_{j=1}^{m_i}(1-r_{ij})^{x_{ij}}] & \\ \text{s.t.} \begin{cases} \sum_{i=1}^{s}\sum_{j=1}^{m_i} c_{ij}x_{ij} \le C_{\max} & (1.1) \\ \sum_{i=1}^{s}\sum_{j=1}^{m_i} w_{ij}x_{ij} \le W_{\max} & (1.2) \\ x_{ij} \in N \qquad i=1,...,s\ ;\ j=1,...,m_i & (1.3) \end{cases} \end{cases}$$

$1-\prod_{j=1}^{m_i}(1-r_{ij})^{x_{ij}}$ is the reliability of subsystem i and $\prod_{i=1}^{s}[1-\prod_{j=1}^{m_i}(1-r_{ij})^{x_{ij}}]$ is the reliability of the complete system. Constraints (1.1) and (1.2) express the cost and weight limits, respectively

3. Literature review

Most of the mathematical approaches used to solve the problem consider a restricted solution domain by only allowing one component choice for each subsystem, or allowing multiple component choices, but allowing no mixing within a subsystem once a component has been selected. Unfortunately, to prohibit component mixing within a subsystem is not realistic in practice. The reader can consult [Kuo and Wan, 2007] for a state-of-the-art of RAP. We give below some precisions on two references which study problem P1 or a problem close to P1. In [Ramirez-Marquez et al., 2004] the authors consider the series-parallel structure presented in the introduction of this paper but their objective is to maximize the minimum subsystem reliability. In this new

formulation the objective function $\max_x \left(\min_i \left(1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right) \right)$ is much easier to handle than the classical objective function and the problem can be exactly formulated by an integer linear program.

Hsieh [Hsieh, 2002] considers the classical redundancy allocation problem and propose a simple linear programming approach to approximate the integer non linear objective function. In order to construct this approximation, he uses the two following inequalities:

$$1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \leq \prod_{i=1}^{s} [1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}}] \leq \left(1 - \frac{1}{n} \sum_{i=1}^{s} \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right)^n$$

His method consists to solve a continuous linear program and then to construct an integer solution by solving a linear 0-1 knapsack problem. The method is particularly fast but the obtained reliability may be slightly far from the optimal reliability. We can add to the works mentioned in [Kuo and Wan, 2007] three references. In [Levitin et al., 1997] the authors consider the series-parallel structure presented in the introduction of this paper to describe an electric power system. The objective function they consider is different from the classical objective function of (P1) since the system reliability is defined as the ability to satisfy consumer demand which is represented as a piecewise cumulative load curve. A genetic algorithm is used to solve the problem. The solution method described in [Zia and Coit, 2005] for P1 is based on the decomposition approach of column generation. In this approach a linear master problem and multiple nonlinear subproblems are iteratively solved. Computational experiments on 33 well-known problems [Nakagawa and Miyazaki, 1981] show that the column generation approach matches or surpasses three existing methods ([Coit and Smith, 1996], [Kultural-Konak et al., 2003] and [Liang and Smith, 2004]) in 25 of 33 cases. In [Onishi et al., 2007] an exact solution method is developed, based on the improved surrogate constraint method proposed by Nakagawa [Nakagawa, 2003]. This method is used to find optimal solutions to the 33 problems devised by [Nakagawa and Miyazaki, 1981]. The idea of the surrogate constraint method is to translate a multidimensional problem into a surrogate dual problem with a single dimension, by an optimal aggregation of the primal constraints. The approach is particularly efficient if there is no duality gap or if the duality gap is small.

The new method presented in this paper is essentially based on two ideas which, to the best of our knowledge, have not been used in previously published methods. The first idea is a discretization of the reliability $R_i(x)$ of each subsystem which allows to write $R_i(x)$ as a linear expression subject to a linear constraint. This expression is a function of a single real variable and of several Boolean variables. The second idea is a precise upper approximation of the natural

logarithm of (C+t) where C is a positive constant and t is a real variable belonging to a small interval $[0, \varepsilon]$. This upper approximation is quadratic because it contains products of a real variable by a Boolean variable. So we propose to linearize it by a classical technique in order to obtain a mixed 0-1 linear program. The optimal solution of this program provides a near-optimal solution of the redundancy allocation problem and a precise upper bound of its optimal value.

4. Solution method

4.1 Rewriting P1 by discretizing the reliability of each subsystem

Consider $R_i(x)$, the reliability of subsystem i :

$$R_i(x) = 1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \qquad (i = 1,..., s)$$

and suppose, without loss of generality, that in an optimal solution x^* of P1, $L \le R_i(x^*) \le 1$ for all subsystems. We can therefore write:

$$R_i(x) = L + \varepsilon \sum_{k=0}^{p} k\, z_{ik} + t_i \qquad (i = 1,..., s)$$

where ε is a real coefficient between 0 and 1, z_{ik} $(i = 1,..., s; k = 1,..., p)$ are Boolean variables such that $\sum_{k=0}^{p} z_{ik} = 1$ $(i = 1,..., s)$, t_i is a real variable such that $0 \le t_i \le \varepsilon$ and $p = \lfloor (1 - L)/\varepsilon \rfloor$. It is easy to verify that for all values $R_i(\tilde{x})$ between L and 1, there exists $\tilde{z}_{ik} \in \{0,1\}$ $(k = 0,..., p)$ and $0 \le \tilde{t}_i \le \varepsilon$ such that $R_i(\tilde{x}) = L + \varepsilon \sum_{k=0}^{p} k\, \tilde{z}_{ik} + \tilde{t}_i$. Conversely, for all values $\tilde{z}_{ik} \in \{0,1\}$ and $0 \le \tilde{t}_i \le \varepsilon$, $R_i(\tilde{x}) = L + \varepsilon \sum_{k=0}^{p} k\, \tilde{z}_{ik} + \tilde{t}_i$ is a value between L and 1. P1 is therefore equivalent to $\mathrm{P2}_{L,\varepsilon}$ which depends on two parameters: a precision ε and a lower bound L on the reliability of each subsystem.

$$(\mathrm{P2}_{L,\varepsilon})\begin{cases}\max \prod_{i=1}^{s}(L+\varepsilon\sum_{k=0}^{p}k\,z_{ik}+t_i) & & & \\ \text{s.t.}\begin{cases}\sum_{k=0}^{p}z_{ik}=1 & i=1,...,s & & (2.1)\\ L+\varepsilon\sum_{k=0}^{p}k\,z_{ik}+t_i\le 1-\prod_{j=1}^{m_i}(1-r_{ij})^{x_{ij}} & i=1,...,s & & (2.2)\\ \sum_{i=1}^{s}\sum_{j=1}^{m_i}c_{ij}x_{ij}\le C_{\max} & & & (2.3)\\ \sum_{i=1}^{s}\sum_{j=1}^{m_i}w_{ij}x_{ij}\le W_{\max} & & & (2.4)\\ x_{ij}\in N & i=1,...,s & j=1,...,m_i & (2.5)\\ z_{ik}\in\{0,1\} & i=1,...,s & k=0,...,p & (2.6)\\ 0\le t_i\le\varepsilon & i=1,...,s & & (2.7)\end{cases}\end{cases}$$

Remark that constraint (2.2) can be considered as an inequality because of the objective function to maximize. In fact, at the optimum of $\mathrm{P2}_{L,\varepsilon}$, both sides of this inequality are equal.

Consider the natural logarithm of the objective function of $\mathrm{P2}_{L,\varepsilon}$ and the natural logarithm of both sides of constraint (2.2) rewritten as $\prod_{j=1}^{m_i}(1-r_{ij})^{x_{ij}}\le 1-L-\varepsilon\sum_{k=0}^{p}k\,z_{ik}-t_i$. Taking into account constraint (2.1) we get program $\mathrm{P3}_{L,\varepsilon}$ equivalent to $\mathrm{P2}_{L,\varepsilon}$:

$$(\mathrm{P3}_{L,\varepsilon})\begin{cases}\max \sum_{i=1}^{s}\sum_{k=0}^{p}[\ln(L+k\varepsilon+t_i)]\,z_{ik} & & & \\ \text{s.t.}\begin{cases}\sum_{k=0}^{p}z_{ik}=1 & i=1,...,s & & (3.1)\\ \sum_{j=1}^{m_i}[\ln(1-r_{ij})]x_{ij}\le\sum_{k=0}^{p}[\ln(1-L-k\varepsilon-t_i)]\,z_{ik} & i=1,...,s & & (3.2)\\ \sum_{i=1}^{s}\sum_{j=1}^{m_i}c_{ij}x_{ij}\le C_{\max} & & & (3.3)\\ \sum_{i=1}^{s}\sum_{j=1}^{m_i}w_{ij}x_{ij}\le W_{\max} & & & (3.4)\\ x_{ij}\in N & i=1,...,s & j=1,...,m_i & (3.5)\\ z_{ik}\in\{0,1\} & i=1,...,s & k=0,...,p & (3.6)\\ 0\le t_i\le\varepsilon & i=1,...,s & & (3.7)\end{cases}\end{cases}$$

Remark. If we fix, in $\mathrm{P3}_{L,\varepsilon}$, $t_i=0$ for all *i*, we obtain an integer linear program. Solving this program amounts to search for an optimal system reliability by measuring the reliability of each stage by values belonging to $\{L, L+\varepsilon, L+2\varepsilon,..., L+p\varepsilon\}$. Moreover if we replace $k\varepsilon$ by $(k+1)\varepsilon$ in the objective function, the optimal value of the obtained program is an upper bound of the optimal

reliability. Indeed $\sum_{i=1}^{s}\sum_{k=0}^{p}[\ln(L+(k+1)\varepsilon)]\,z_{ik} \geq \sum_{i=1}^{s}\sum_{k=0}^{p}[\ln(L+k\varepsilon+t_i)]\,z_{ik}$ for all possible values of t_i and the constraint $\sum_{j=1}^{m_i}[\ln(1-r_{ij})]x_{ij} \leq \sum_{k=0}^{p}[\ln(1-L-k\varepsilon)]\,z_{ik}$ is a relaxation of constraint (3.2). In the following, we propose a tighter overestimation of the objective function and a tighter relaxation of constraint (3.2).

4.2 A relaxation of $\mathbf{P3}_{L,\varepsilon}$

It is known that $\ln(L+k\varepsilon)+t_i/(L+k\varepsilon)$ is an upper approximation of $\ln(L+k\varepsilon+t_i)$, if $L+k\varepsilon>0$. Now consider a more precise upper approximation of $\ln(L+k\varepsilon+t_i)$. For all values of u such that $0 \leq u \leq 1$ the following inequality holds :

$$\ln(L+k\varepsilon+t_i) = \ln(L+k\varepsilon+u\varepsilon+t_i-u\varepsilon) \leq \ln(L+k\varepsilon+u\varepsilon)+(t_i-u\varepsilon)/(L+k\varepsilon+u\varepsilon) \quad (1)$$

Let $U=\{u_1,u_2,...u_{q_1}\}$ be a set of q real numbers such that $0 \leq u_1 < u_2 < ... < u_{q_1} \leq 1$. Using (1), $\min_{l=1,...,q_1}\{\ln(L+k\varepsilon+u_l\varepsilon)+(t_i-u_l\varepsilon)/(L+k\varepsilon+u_l\varepsilon)\}$ is an upper approximation of $\ln(L+k\varepsilon+t_i)$. Obviously, if $u_1=0$, this minimum is less than or equal to $\ln(L+k\varepsilon)+t_i(L+k\varepsilon)$.

Let us see a numerical example of this approximation. Let $L=0.7$, $k=26$, $\varepsilon=0.01$, $t_i=0.0085$ and $U=\{0,0.2,0.4,0.6,0.8,1\}$. We get

$$\ln(L+k\varepsilon+t_i) \approx \text{-}0.0320068,$$

$$\ln(L+k\varepsilon)+\frac{t_i}{L+k\varepsilon} \approx \text{-}0.0319678$$

and $$\min_{l=1,...,q_1}\{\ln(L+k\varepsilon+u_l\varepsilon)+(t_i-u_l\varepsilon)/(L+k\varepsilon+u_l\varepsilon)\} \approx \text{-}0.0320067.$$

In the same way, if $V=\{v_1,v_2,...v_{q_2}\}$ is a set of q real numbers such that $0=v_1<v_2<...<v_{q_2}\leq 1$ and $1-L-k\varepsilon-v_{q_2}\varepsilon>0$, an upper approximation of $\ln(1-L-k\varepsilon-t_i)$ is given by $\min_{l=1,...,q_2}\{\ln(1-L-k\varepsilon-v_i\varepsilon)-(t_i-v_i\varepsilon)/(1-L-k\varepsilon-v_i\varepsilon)\}$.

We can now construct $\mathrm{P4}_{L,\varepsilon,U,V}$, the relaxation of $\mathrm{P3}_{L,\varepsilon}$.

Notation : let $f_l(L,k,\varepsilon,U,t_i) = \ln(L+k\varepsilon+u_l\varepsilon)+(t_i-u_l\varepsilon)/(L+k\varepsilon+u_l\varepsilon)$ and $g_l(L,k,\varepsilon,V,t_i) = \ln(1-L-k\varepsilon-v_l\varepsilon)-(t_i-v_l\varepsilon)/(1-L-k\varepsilon-v_l\varepsilon)$.

$$(\mathrm{P4}_{L,\varepsilon,U,V})\begin{cases}\max \sum_{i=1}^{s} \min\limits_{l=1,\ldots,q_1} \sum_{k=0}^{p} f_l(L,k,\varepsilon,U,t_i) z_{ik} & & & \\ \text{s.t.} \begin{cases} \sum_{k=0}^{p} z_{ik} = 1 & i=1,\ldots,s & & (4.1) \\ \sum_{j=1}^{m_i} [\ln(1-r_{ij})] x_{ij} \le \min\limits_{l=1,\ldots,q_2} \sum_{k=0}^{p} g_l(L,k,\varepsilon,V,t_i) z_{ik} & i=1,\ldots,s & & (4.2) \\ \sum_{i=1}^{s} \sum_{j=1}^{m_i} c_{ij} x_{ij} \le C_{\max} & & & (4.3) \\ \sum_{i=1}^{s} \sum_{j=1}^{m_i} w_{ij} x_{ij} \le W_{\max} & & & (4.4) \\ x_{ij} \in N & i=1,\ldots,s & j=1,\ldots,m_i & (4.5) \\ z_{ik} \in \{0,1\} & i=1,\ldots,s & k=0,\ldots,p & (4.6) \\ 0 \le t_i \le \varepsilon & i=1,\ldots,s & & (4.7) \end{cases} \end{cases}$$

4.3. Linearization of $\mathrm{P4}_{L,\varepsilon,U,V}$

Program $\mathrm{P4}_{L,\varepsilon,U,V}$ is not linear. Indeed it contains the nonlinear expressions $\min\limits_{l=1,\ldots,q_1} \sum_{k=0}^{p} f_l(L,k,\varepsilon,U,t_i) z_{ik}$ in the objective function and $\min\limits_{l=1,\ldots,q_2} \sum_{k=0}^{p} g_l(L,k,\varepsilon,V,t_i) z_{ik}$ in constraints (4.2). Classically, $\mathrm{P4}_{L,\varepsilon,U,V}$ is equivalent to $\mathrm{P5}_{L,\varepsilon,U,V}$:

$$(\mathrm{P5}_{L,\varepsilon,U,V})\begin{cases}\max \sum_{i=1}^{s} \alpha_i & & & & \\ \text{s.t.} \begin{cases} \sum_{k=0}^{p} z_{ik} = 1 & i=1,\ldots,s & & & (5.1) \\ \alpha_i \le \sum_{k=0}^{p} f_l(L,k,\varepsilon,U,t_i) z_{ik} & i=1,\ldots,s & l=1,\ldots,q_1 & & (5.2) \\ \beta_i \le \sum_{k=0}^{p} g_l(L,k,\varepsilon,V,t_i) z_{ik} & i=1,\ldots,s & l=1,\ldots,q_2 & & (5.3) \\ \sum_{j=1}^{m_i} [\ln(1-r_{ij})] x_{ij} \le \beta_i & i=1,\ldots,s & & & (5.4) \\ \sum_{i=1}^{s} \sum_{j=1}^{m_i} c_{ij} x_{ij} \le C_{\max} & & & & (5.5) \\ \sum_{i=1}^{s} \sum_{j=1}^{m_i} w_{ij} x_{ij} \le W_{\max} & & & & (5.6) \\ x_{ij} \in N & i=1,\ldots,s & j=1,\ldots,m_i & & (5.7) \\ z_{ik} \in \{0,1\} & i=1,\ldots,s & k=0,\ldots,p & & (5.8) \\ 0 \le t_i \le \varepsilon & i=1,\ldots,s & & & (5.9) \end{cases} \end{cases}$$

$(\mathrm{P5}_{L,\varepsilon,U,V})$ is not yet linear since constraints (5.2) and (5.3) contain the quadratic terms $t_i z_{ik}$ $(i=1,\ldots,s; 0=1,\ldots,p)$. Recall that t_i is a real variable such that $0 \le t_i \le \varepsilon$ and z_{ik}, a Boolean variable. In order to linearize the product $t_i z_{ik}$ we substitute y_{ik} for this product and add the four following linearization constraints : $y_{ik} \le z_{ik}\varepsilon$, $y_{ik} \le t_i$, $y_{ik} \ge t_i - \varepsilon(1-z_{ik})$ and $y_{ik} \ge 0$. By examining successively the two possible values of z_{ik} we immediately see that $y_{ik} = t_i z_{ik}$ if and

only if these four constraints are satisfied. By using this linearization technique we obtain the mixed integer linear program $(P6_{L,\varepsilon,U,V})$ equivalent to $(P5_{L,\varepsilon,U,V})$.

$$
(P6_{L,\varepsilon,U,V})\begin{cases}\max \sum_{i=1}^{s}\alpha_i & & & \\ \text{s.t.} & \sum_{k=0}^{p} z_{ik} = 1 & i=1,...,s & & (6.1)\\ & \alpha_i \le \sum_{k=0}^{p} [\ln(L+k\varepsilon+u_l\varepsilon)z_{ik} & & & (6.2)\\ & +(y_{ik}-u_l\varepsilon\, z_{ik})/(L+k\varepsilon+u_l\varepsilon)] & i=1,...,s & l=1,...,q_1 & (6.3)\\ & \beta_i \le \sum_{k=0}^{p}[\ln(1-L-k\varepsilon-v_l\varepsilon)z_{ik} & & & (6.4)\\ & -(y_{ik}-v_l\varepsilon\, z_{ik})/(1-L-k\varepsilon-v_l\varepsilon)] & i=1,...,s & l=1,...,q_2 & (6.5)\\ & \sum_{j=1}^{m_i}\ln(1-r_{ij})x_{ij} \le \beta_i & i=1,...,s & & (6.6)\\ & \sum_{i=1}^{s}\sum_{j=1}^{m_i} c_{ij}x_{ij} \le C_{\max} & & & (6.7)\\ & \sum_{i=1}^{s}\sum_{j=1}^{m_i} w_{ij}x_{ij} \le W_{\max} & & & (6.8)\\ & y_{ik} \le \varepsilon\, z_{ik},\ y_{ik} \le t_i,\ y_{ik} \ge t_i - \varepsilon(1-z_{ik}) & i=1,...,s & k=0,...,p & (6.9)\\ & y_{ik} \ge 0,\ z_{ik} \in \{0,1\} & i=1,...,s & k=0,...,p & (6.10)\\ & x_{ij} \in N & i=1,...,s & j=1,...,m_i & (6.11)\\ & 0 \le t_i \le \varepsilon & i=1,...,s & & (6.12)\end{cases}
$$

Let $(x^*, y^*, z^*, t^*, \alpha^*, \beta^*)$ be an optimal solution of $P6_{L,\varepsilon,U,V}$. Its value $\sum_{i=1}^{s}\alpha_i^*$ is an upper bound of the optimal value of the initial problem and x_{ij}^* is a feasible solution of the initial problem. The corresponding reliability of the system is equal to $\prod_{i=1}^{s}[1-\prod_{j=1}^{m_i}(1-r_{ij})^{x_{ij}^*}]$. In other words, an optimal solution of the mixed integer linear program $P6_{L,\varepsilon,U,V}$ defines a configuration of the system (x_{ij}^*); this system has a certain reliability, $\prod_{i=1}^{s}[1-\prod_{j=1}^{m_i}(1-r_{ij})^{x_{ij}^*}]$, and the reliability of the best possible system is less than or equal to the optimal value of $P6_{L,\varepsilon,U,V}$. If ε is sufficiently small and the cardinal of *U* and *V* sufficiently large the optimal solution of $P6_{L,\varepsilon,U,V}$ will be close to the optimal solution of P1. However, the smaller is ε, the larger is the number of variables z_{ik}, and the larger are the cardinals of *U* and *V*, the larger is the number of constraints.

4.4. Solution of RAP by iterative solution of $P6_{L,\varepsilon,U,V}$

We propose a solution method for RAP based on iterative solutions of the mixed integer linear program $P6_{L,\varepsilon,U,V}$. At each iteration, we try to increase the value of *L* and we decrease the value of ε. By this iterative method we try to maintain the number of variables z_{ik} in $P6_{L,\varepsilon,U,V}$ at

an acceptable level. Note that if the optimal system reliability is greater than or equal to *R*, then, obviously, the reliability of all subsystems in an optimal solution is also greater than or equal to *R*. One can also remark that the reliability of all subsystems is greater than or equal to $r^{\min} = \min\{r_{ij} : i = 1,...,s;\ j = 1,...,m_i\}$. ε^0 is the chosen initial precision and at each step of the while loop the precision is multiplied by ρ, a real parameter such that $0 < \rho < 1$. Let $D \geq 1$ be an integer number. At each iteration, for a given value of ε we define *U* and *V* as follows : $u_l = (l-1)\varepsilon / D$ and $v_l = (l-1)\varepsilon / D$ $(l = 1,...,D)$.

Algorithm $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$
Let R^0 be a feasible reliability of the system
Let ε^0 be an initial precision
$L \leftarrow \max\{R^0, r^{\min}\}$; $\varepsilon \leftarrow \varepsilon^0$
While $\varepsilon \geq \varepsilon^{\min}$ do
Solve $P6_{L,\varepsilon,U,V}$
Let R^* be the corresponding system reliability
$L \leftarrow \max\{R^*, r^{\min}\}$
$\varepsilon \leftarrow \rho\varepsilon$
end do

The near-optimal solution is given by the solution of $P6_{L,\varepsilon,U,V}$ at the last iteration of the algorithm (by the values of the variables *x*); the upper bound on an optimal reliability is given by the value of this solution. The parameters $L, \varepsilon^0, \rho, \varepsilon^{\min}$ and *D* will be chosen empirically, taking into account the difficulty of the problem and after some experiments. Note that an initial feasible reliability of the complete system can be determined, for example, by solving $P6_{L,\varepsilon,U,V}$ with $L = r_{\min}$, $\varepsilon = 0.1$ or 0.01, $U = \{0\}$ and $V = \{0\}$.

5. Computational experiments

The different instances of $P6_{L,\varepsilon,U,V}$ have been solved using the solver CPLEX 8 [CPLEX, 2002] and the modeling language AMPL [Fourer et al., 1993]. The experiments have been carried out on a PC Pentium 4 with a 1.8 GHz processor.

5.1 First computational tests: a gearbox reliability optimization problem [Zhao et al., 2005]

The first test problem used here to demonstrate the efficiency of *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$ for solving RAP is a gearbox reliability optimization problem presented in [Zhao et al., 2005]. In this reference, in order to apply their method, the authors assumed, for all stages, that the minimal

number of components is equal to 2 and that the maximal number of components is equal to 5. We have noticed that, with our approach, identical solutions are obtained with or without this restriction. Results using an ant colony system (ACS) meta-heuristic [Zhao et al., 2005] and our integer linear programming approach (ILP) are presented in Tables 1a and 1b. In the experiments we choose the following values of the parameters in *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$: $L = r^{\min} = 0.7$, $\varepsilon^0 = 0.1$, $\rho = 0.1$, $\varepsilon^{\min} = 0.001$, and $D = 51$. As results indicate, ILP outperforms ACS. In 16 problem cases, ILP obtained better system reliability and in 3 problem cases ILP and ACS obtained identical system reliability. Let us note that ILP is also efficient from the computational time point of view since a total of about 5 seconds were necessary to solve the 19 problems.

Table 1a. Results obtained for a gearbox reliability optimization problem devised by [Zhao et al., 2005] ; comparison of two approaches : ACS [Zhao et al., 2005] and our ILP approach

No.	Maximal cost	Maximal weight	Reliability [Zhao et al., 2005] R_{ZLD}	Reliability Our approach R_{ILP}
1	40	115	0.9861	0.98629
2	55	125	0.9973	0.99758
3	65	130	0.9977	0.99857
4	60	120	0.9968	0.99699
5	60	130	0.9977	0.99794
6	60	140	0.9985	0.99855
7	60	150	0.9987	0.99878
8	65	120	0.9968	0.99744
9	65	130	0.9977	0.99857
10	65	140	0.9988	0.99905
11	65	150	0.9990	0.99905
12	70	120	0.9968	0.99792
13	70	130	0.9988	0.99881
14	70	140	0.9990	0.99905
15	70	150	0.9992	0.99949
16	75	120	0.9968	0.99792
17	75	130	0.9988	0.99881
18	75	140	0.9992	0.99925
19	75	150	0.9995	0.99949

Table 1b. Summarized results of the experiments concerning the 19 instances devised by [Zhao et al., 2005]

	improvement	gap	CPU Time (seconds)
min	0.0000	$4.1\ 10^{-7}$	0.09
Av.	$3.5\ 10^{-4}$	$9.3\ 10^{-7}$	0.27
max	0.0011	$1.8\ 10^{-6}$	1.08

# $R_{ILP} > R_{ZLD}$	# $R_{ILP} = R_{ZLD}$	# $R_{ILP} < R_{ZLD}$
16	3	0

In Table 1b *improvement* is equal to the difference $R_{ILP} - R_{ZLD}$, *gap* is equal to the optimal value provided by *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$ minus R_{ILP}. For example, on the average, the gap between the optimal reliability and the reliability obtained by our method is less than or equal to 9.3 10^{-7}. $\#R_{ILP} > R_{ZLP}$ (reps. $\#R_{ILP} = R_{ZLP}$, $\#R_{ILP} < R_{ZLP}$) gives the number of problems for which $R_{ILP} > R_{ZLP}$ (reps. $R_{ILP} = R_{ZLP}$, $R_{ILP} < R_{ZLP}$).

Results presented in Table 1 show that the ILP approach allows to obtain the optimal reliability for the 19 problems with a precision of at least 1.8 10^{-6} since for all these problems the value of *gap* is less than or equal to 1.8 10^{-6}.

5.2 Second computational tests : instances proposed in [Nakagawa and Miyazaki, 1981]

The second test problem used to demonstrate the efficiency of *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$ was originally proposed by [Fyffe et al., 1968]. They specified 130 units of system cost, 170 units of system weight and minimum redundancy level $p_i = 1$ $(i = 1,...,s)$. Nakagawa and Miyazaki [Nakagawa and Miyazaki, 1981] devised 33 variations of the original problem. They fixed the cost constraint $C_{\max} = 130$ and the weight constraint $W_{\max}$ varies from 159 to 191. Only identical components are used in both papers. Many authors considered this problem without restricting the component mixing. Table 2 presents the optimal reliability for the 33 problems in this latter case, the references that permitted to obtain this reliability and the solution obtained by ILP approach. Note that for the 33 problems the reliability is proved to be optimal only in [Onishi et al., 2007]. We also give the CPU time required to determine this reliability and the corresponding upper bound, i.e. the optimal value of $P6_{L,\varepsilon,U,V}$ at the end of the execution of *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$. For example, for problem no.18, the optimal reliability is 0.9749. It has been obtained in the following references : [Liang and Chen, 2007], [Liang and Smith, 2004], [Onishi et al., 2007], [You and Chen, 2005], and [Zia and Coit, 2005]. By our approach we obtained the same system reliability within 2.5 seconds of CPU time. Table 3 gives the references we considered for the solution of this problem. It also presents the abbreviations used in Table 2 and the kind of method used in each reference. In the experiments we choose the following values of the parameters in *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$: $L = r^{\min} = 0.7$, $\varepsilon^0 = 0.1$, $\rho = 0.1$, $\varepsilon^{\min} = 0.001$, and $D = 6$.

Table 2a. Comparison of ILP results with best available ones
for the 33 instances devised by [Nakagawa and Miyazaki, 1981]

No.	Maximal weight	optimal reliability R*	References	ILP Approach R_{ILP}	CPU time (seconds)
1	191	0.9868	CY KSC LC LS OKJN YC ZC ZLD	0.9868	35.80
2	190	0.9864	CY KSC LC OKJN YC ZC ZLD	0.9864	27.69
3	189	0.9859	CY KSC LC OKJN YC ZLD	0.9859	8.63
4	188	0.9854	KSC OKJN YC ZLD	0.9854	14.86
5	187	0.9847	KSC LC LS OKJN YC ZC ZLD	0.9847	19.41
6	186	0.9842	CY KSC LC OKJN YC ZC ZLD	0.9842	1.11
7	185	0.9835	CK KSC LC LS OKJN YC ZLD	0.9835	5.04
8	184	0.9830	KSC LC LS OKJN YC ZC	0.9830	6.69
9	183	0.9823	KSC LC OKJN YC	0.9823	17.43
10	182	0.9815	CY KSC LC LS OKJN YC	0.9815	8.12
11	181	0.9810	CY KSC LC OKJN YC ZC ZLD	0.9810	9.13
12	180	0.9803	CY KSC LC LS OKJN YC ZC ZLD	0.9803	7.71
13	179	0.9795	CY KSC LC LS OKJN YC ZC ZLD	0.9795	4.99
14	178	0.9784	KSC LC LS OKJN YC ZC ZLD	0.9784	7.93
15	177	0.9776	LC LS OKJN YC ZC ZLD	0.9776	6.43
16	176	0.9767	CY KSC LC OKJN YC ZLD	0.9767	12.93
17	175	0.9757	CY KSC LC LS OKJN YC	0.9757	4.20
18	174	0.9749	LC LS OKJN YC ZC	0.9749	2.50
19	173	0.9738	CY KSC LC LS OKJN YC ZC	0.9738	2.62
20	172	0.9730	CY KSC LC LS OKJN YC ZC	0.9730	3.78
21	171	0.9719	CS CY KSC LC LS OKJN YC ZC	0.9719	9.52
22	170	0.9708	CS CY KSC LC LS OKJN YC ZC	0.9708	7.45
23	169	0.9693	CY KSC LC LS OKJN YC ZC	0.9693	13.11
24	168	0.9681	CS CY KSC LC LS OKJN YC ZC	0.9681	15.77
25	167	0.9663	CK CS CY KSC LC LS OKJN YC ZC ZLD	0.9663	22.35
26	166	0.9650	CS CY KSC LC LS OKJN YC ZC	0.9650	15.49
27	165	0.9637	CS CY KSC LC LS OKJN YC ZC	0.9637	26.22
28	164	0.9624	CS CY KSC LC LS OKJN YC ZC ZLD	0.9624	27.32
29	163	0.9606	CK CS CY LC LS OKJN YC ZC ZLD	0.9606	13.61
30	162	0.9592	CY LC LS OKJN YC ZC ZLD	0.9592	15.61
31	161	0.9580	CS CY LC LS OKJN YC ZC ZLD	0.9580	12.06
32	160	0.9557	CS CY LC LS OKJN YC ZC ZLD	0.9557	35.47
33	159	0.9546	CY LC LS OKJN YC ZC ZLD	0.9546	17.23

Table 2.b. Summarized results of the experiments concerning
The 33 instances devised by [Nakagawa and Miyazaki, 1981]

	improvement	gap	CPU Time (seconds)
min	-	$4.4\ 10^{-5}$	1.11
Av.	-	$6.0\ 10^{-5}$	13.28
max	-	$8.2\ 10^{-5}$	35.80

# $R_{ILP} > R^*$	# $R_{ILP} = R^*$	# $R_{ILP} < R^*$
-	33	0

Table 2 shows that by our ILP approach we obtained the optimal reliability for all the problems. For the 33 problems the mean CPU time is equal to 1.11 seconds and the gap between the obtained reliability and the corresponding upper bound is always less than $8.2\ 10^{-5}$.

Table 3. Some references and type of methods considering the 33 instances of the redundancy allocation problem devised by [Nakagawa and Miyazaki, 1981]

Abbreviation	Reference	Type of method
CK	[Coit and Konak, 2006]	heuristic based on solving a sequence of linear programming problems
CS	[Coit and Smith, 1996]	genetic algorithm
CY	[Chen and You, 2005]	Immune algorithms-based approach
H	[Hsieh, 2002]	approximation by linear programming
KSC	[Kulturel-Konak et al., 2003]	tabu search meta-heuristic
LC	[Liang and Chen, 2007]	variable neighborhood search algorithm
LS	[Liang and Smith, 2004]	ant colony optimization
OKJN	[Onishi et al., 2007]	exact solution method, based on the improved surrogate constraint (ISC) method
YC	[You and Chen, 2005]	greedy method and genetic algorithm
ZC	[Zia and Coit, 2005]	decomposition approach of column generation
ZLD	[Zhao et al., 2005]	ant colony system meta-heuristic

5.3 Third computational tests : instances proposed in [Coit and Konak, 2006]

The previous problem devised by [Nakagawa and Miyazaki, 1981] is a well-known test problem and it has been extensively used in the literature to evaluate alternative approaches for RAP. However, as mentioned in [Coit and Konak, 2006] it has some drawbacks as a test problem because some of component options clearly dominate others, i.e. they have lower cost and weight, but higher reliability than others. These authors claim that many component options can be eliminated if the problem data are carefully analyzed and propose three new problems with difficult tradeoffs between components. Each problem has 20 subsystems with four available component options and the maximum number of components allowed in each subsystem is 8. For these three problems 36 different combinations of cost and weight constraint limits are defined. The results obtained by our approach on these three problems are presented in Table 4 together with the result obtained by Coit and Konak by their multiple weighted objective heuristic. We have noticed in our experiments that as expected these 3 problems are much more difficult than the problem devised by [Nakagawa and Miyazaki, 1981] and that, as for Coit and Konak, Problem 3 is the most challenging problem among them.

Taking into account the difficulty of these three problems we choose in the experiments the following values of the parameters in *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$ which correspond to a unique iteration of the while loop.:

Problem 2 : $L = r^{\min} = 0.7$, $\varepsilon^0 = 0.05$, $\rho = 0.1$, $\varepsilon^{\min} = 0.05$, and $D = 21$.
Problem 3: $L = r^{\min} = 0.9$, $\varepsilon^0 = 0.05$, $\rho = 0.1$, $\varepsilon^{\min} = 0.05$, and $D = 21$.
Problem 4: $L = r^{\min} = 0.8$, $\varepsilon^0 = 0.05$, $\rho = 0.1$, $\varepsilon^{\min} = 0.05$, and $D = 21$.

Table 4. Computational experiments for the 3x36 instances presented in [Coit and Konak, 2006]; comparison of two approaches : the MWO heuristic [Coit and Konak, 2006] and our ILP approach
For each problem the CPU time is limited to 300 seconds
* An optimal solution of $P6_{L,\varepsilon,U,V}$ has not been found within 300 seconds

			[Coit and Konak, 2006] **Problem 2**				[Coit and Konak, 2006] **Problem 3**				[Coit and Konak, 2006] **Problem 4**			
	C	W	R_{CK}	R_{ILP}	Upper bound ILP	CPU ILP	R_{CK}	R_{ILP}	Upper bound ILP	CPU ILP	R_{CK}	R_{ILP}	Upper bound ILP	CPU ILP
1	100	100	0.05268	0.06269	0.06270	101.89	0.17265	0.17265	0.17265	0.12	0.14722	0.15085*	0.20454*	300.00*
2	100	130	0.07702	0.08592	0.08592	119.33	0.25035	0.25287	0.25290	3.58	0.32840	0.32607*	0.39673*	300.00*
3	100	160	0.07702	0.08592	0.08592	26.80	0.31659	0.31848*	0.32328*	300.00*	0.55503	0.56460	0.56474	11.33
4	100	190	0.07702	0.08592	0.08592	26.92	0.39429	0.39833*	0.40270*	300.00*	0.64773	0.65377	0.65509	1.23
5	100	220	0.07702	0.08592	0.08592	26.83	0.48209	0.48156*	0.49803*	300.00*	0.72987	0.73348	0.73377	1.54
6	100	250	0.07702	0.08592	0.08592	26.81	0.58655	0.58875*	0.60130*	300.00*	0.76426	0.78414	0.78452	0.49
7	130	100	0.08091	0.08091	0.08091	15.17	0.24129	0.24388	0.24389	8.04	0.31355	0.31311*	0.36822*	300.00*
8	130	130	0.26884	0.27280	0.27282	12.77	0.31593	0.32242	0.32247	14.33	0.56078	0.56627	0.56641	12.20
9	130	160	0.31506	0.33524	0.33526	2.53	0.39476	0.40543	0.40551	17.18	0.66594	0.67087	0.67112	0.68
10	130	190	0.31061	0.33524	0.33526	1.84	0.48277	0.50330	0.50342	24.64	0.74961	0.75992	0.76031	0.30
11	130	220	0.31470	0.33524	0.33526	1.84	0.59041	0.61452	0.61471	33.62	0.83421	0.83836	0.83876	8.44
12	130	250	0.31061	0.33524	0.33526	1.84	0.71971	0.71863*	0.76459*	300.00*	0.89123	0.89274	0.89408	0.43
13	160	100	0.08091	0.08091	0.08091	15.21	0.30389	0.30954	0.30959	43.24	0.53033	0.53498	0.53505	1.88
14	160	130	0.28595	0.28824	0.28826	6.32	0.38961	0.40612	0.40624	2.95	0.64802	0.65427	0.65452	4.52
15	160	160	0.52941	0.55183*	0.58938*	300.00*	0.48297	0.50632	0.50646	13.64	0.74884	0.75725	0.75751	4.41
16	160	190	0.65762	0.65762	0.65766	0.56	0.59047	0.61553*	0.65847*	300.00*	0.82922	0.84001	0.84035	77.22
17	160	220	0.65762	0.65762	0.65766	0.35	0.75473	0.75740*	0.76242*	300.00*	0.89751	0.89970	0.90085	1.24
18	160	250	0.65762	0.65762	0.65766	0.36	0.88177	0.88302	0.88345	0.86	0.92416	0.92612	0.92963	3.00
19	190	100	0.08091	0.08091	0.08091	15.18	0.38067	0.38217	0.38230	130.95	0.60364	0.61204	0.61216	2.83
20	190	130	0.28595	0.28824	0.28826	6.33	0.48074	0.50168	0.50191	1.94	0.72425	0.73550	0.73574	3.41
21	190	160	0.60371	0.61592	0.61596	14.03	0.61995	0.62126	0.62155	10.49	0.82697	0.83434	0.83474	1.00
22	190	190	0.75874	0.76866	0.76893	1.31	0.75809	0.76046*	0.77413*	300.00*	0.89718	0.89833	0.89945	1.06
23	190	220	0.79847	0.82049	0.82084	0.31	0.88293	0.88400	0.88453	0.84	0.92675	0.92686	0.93119	5.00
24	190	250	0.80097	0.82049	0.82084	0.27	0.90783	0.90847	0.91111	0.75	0.94725	0.94981	0.95730	1.10
25	220	100	0.08091	0.08091	0.08091	15.13	0.46579	0.46989	0.47016	28.85	0.63909	0.66592	0.66610	0.31
26	220	130	0.28595	0.28824	0.28826	6.32	0.58585	0.60726	0.60749	3.17	0.79882	0.80449	0.80479	3.02
27	220	160	0.60371	0.61592	0.61596	19.96	0.72885	0.75474*	0.78328*	300.00*	0.88788	0.89026	0.89118	0.46
28	220	190	0.79824	0.80807	0.80853	0.58	0.88214	0.88214	0.88275	1.06	0.92239	0.92301	0.92705	0.31
29	220	220	0.87281	0.87583	0.87643	1.16	0.90616	0.90653	0.90910	10.26	0.94678	0.94790	0.95542	21.31
30	220	250	0.89454	0.90006	0.90064	0.32	0.92302	0.92525*	0.93095*	300.00*	0.96779	0.96805	0.97857	2.09
31	250	100	0.08091	0.08091	0.08091	15.15	0.54857	0.57507	0.57546	1.04	0.63909	0.66592	0.66610	0.31
32	250	130	0.28595	0.28824	0.28826	6.32	0.71172	0.71129*	0.73832*	300.00*	0.84744	0.85793	0.85845	0.96
33	250	160	0.60371	0.61592	0.61596	19.89	0.87444	0.87629	0.87691	0.34	0.91345	0.91481	0.91743	0.96
34	250	190	0.79824	0.80807	0.80853	0.58	0.90171	0.90389	0.90648	4.21	0.94285	0.94406	0.95006	0.52
35	250	220	0.88581	0.89061	0.89142	0.98	0.92388	0.92364	0.92896	64.31	0.96401	0.96585	0.97613	1.90
36	250	250	0.92957	0.93069	0.93225	0.6	0.94080	0.94097	0.94791	277.33	0.98031	0.98007	0.99421	3.18

In Table 4 "*" means that the optimal solution of $P6_{L,\varepsilon,U,V}$ has not been found within 300 seconds of CPU time. However, a feasible solution has always been found within this time and the corresponding reliability is indicated in the table. We also indicate the upper bound provided by CPLEX after 300 seconds of computation.

Table 4b. Summarized results of the experiments concerning the 36 instances of Problem 2 devised by [Coit and Konak, 2006]

	improvement	gap	CPU Time (seconds)
min	0.00000	0.00000	0.27
Av.	0.00840	0.00121	22.55
max	0.02463	0.03755	300.00

# $R_{ILP} > R_{ZLD}$	# $R_{ILP} = R_{ZLD}$	# $R_{ILP} < R_{ZLD}$
28	8	0

Table 4c. Summarized results of the experiments concerning the 36 instances of Problem 3 devised by [Coit and Konak, 2006]

	improvement	gap	CPU Time (seconds)
min	-0.00108	0.00000	0.12
Av.	0.02650	0.00643	111.05
max	0.02463	0.04596	300.00

# $R_{ILP} > R_{ZLD}$	# $R_{ILP} = R_{ZLD}$	# $R_{ILP} < R_{ZLD}$
30	2	4

Table 4d. Summarized results of the experiments concerning the 36 instances of Problem 4 devised by [Coit and Konak, 2006]

	improvement	gap	CPU Time (seconds)
min	-0.00233	0.00007	0.30
Av.	0.00583	0.00724	29.96
max	0.02683	0.07066	300.00

# $R_{ILP} > R_{ZLD}$	# $R_{ILP} = R_{ZLD}$	# $R_{ILP} < R_{ZLD}$
34	0	2

The results presented in Tables 4a, 4b, 4c and 4d show that the three problem devised by [Coit and Konak, 2006] are much more difficult - at least for our approach - than the problem devised by [Nakagawa and Miyazaki, 1981] (see Table 2a and 2b). However, our integer linear programming approach allows obtaining a reliability better than that obtained by [Coit and Konak, 2006] in 92 cases over 108. The computation times remain relatively small except

in 15 cases where CPLEX cannot find the optimal solution of $P6_{L,\varepsilon,U,V}$ within 300 seconds of CPU time. As suggested by [Coit and Konak, 2006] the three problems are difficult because of the combination of the three types of data : cost, weight and reliability. In order to illustrate that the difficulty comes mainly from this point, we have considered problem 3 without weight constraint. In this case the problem becomes very easy to solve. We give in Table 5 the computational results obtained by *Algorithm* $(L, \varepsilon^0, \rho, \varepsilon^{\min}, D)$ with the following parameters : $L = r_{\min} = 0.9$, $\varepsilon^0 = 0.01$, $\rho = 0.1$, $\varepsilon^{\min} = 0.001$, and $D = 51$.

Table 5. Computational results for 6 instances of problem 3 of [Coit and Konak, 2006] but without constraint weight

	C	R_{ILP}	Upper bound ILP	CPU ILP seconds
1	100	0.993274	0.993278	1.62
2	130	0.999406	0.999434	1.12
3	160	0.999946	0.999971	1.52
4	190	0.999995	0.999998	1.12
5	220	0.999998	1	1.63
6	250	0.999997	1	1.61

The results presented in Table 5 show that the problem 3 of [Coit and Konak, 2006] is not difficult for our approach when we only keep the cost constraint: the mean CPU time required to solve the six problems of Table 5 is equal to 1.44 seconds and the mean gap is equal to 1.1 10^{-5}.

6. Conclusions

We have proposed in this paper to solve the redundant allocation problem by using an integer linear programming software. The experimental results show the effectiveness of the approach compared to previous results published in the literature. Compared to former ones, our work is of three principal interests: first of all the approach is relatively easy to understand and above all uses exclusively a standard, commercially available software. An important asset of the method is its simplicity of implementation in one has an integer programming software. The task becomes really easy if one has, in addition, the use of a tool of modeling like AMPL [Fourer et al., 1993]. Secondly we obtain tight approximate solutions together with a very good performance guarantee. Lastly, as it is well-known with an integer linear programming formulation of the problem, it is possible to add new constraints easily – if they are linear – which is not always the case when a specific algorithm was worked out.

References

T.C. Chen and P.S. You, "Immune algorithms-based approach for redundant reliability problems with multiple component choices", *Computers in Industry*, 56 (2), 195-205, 2005.

D.W. Coit and A. Konak, "Multiple weighted objectives heuristic for the redundancy allocation problem", *IEEE Transactions on Reliability*, 55 (3), 551-558, 2006.

D.W. Coit and A.E. Smith, "Reliability optimization of series-parallel systems using a genetic algorithm", *IEEE Transactions on Reliability*, 45 (2), 254-260, 1996.

CPLEX, ILOG CPLEX 8.0 Reference Manual. ILOG CPLEX Division, Gentilly, France, 2002.

R. Fourer, D.M. Gay and B.W. Kernighan, "AMPL a modeling language for mathematical programming", The Scientific Press Series,1993.

D.E. Fyffe, W.W. Hines and N.K. Lee, "System reliability allocation and a computational algorithm", *IEEE Transactions on Reliability*, 17 (2), 64-69, 1968

Y.-C. Hsieh, "A linear approximation for redundant reliability problems with multiple component choices", *Computer and Industrial Engineering*, 44, 91-103, 2002.

S. Kulturel-Konak, A.E. Smith and D.W. Coit, "Efficiently solving the redundancy allocation problem using tabu search", *IIE Transactions*, 35 (6), 515-526, 2003.

W. Kuo and R. Wan, "Recent advances in optimal reliability allocation", *Computational Intelligence in Reliability Engineering* (SCI), 39, 1-36, 2007.

G. Levitin, A. Lisnianski, and D. Elmakis, “Structure optimization of power system with different redundant elements”, *Electric Power Systems Research*, 43, 19–27, 1997.

Y.-C. Liang and Yi-Ching Chen, "Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm", *Reliability Engineering and System Safety*, 92 (3), 323-331, 2007.

Y.-C. Liang and A.E. Smith, "An ant colony optimization algorithm for the redundancy allocation problem (RAP)", *IEEE Transactions on Reliability*, 53(3), 417-423, 2004.

Y. Nakagawa and S. Miyazaki, "Surrogate constraints algorithm for reliability optimization problems with two constraints", *IEEE Transactions on Reliability*, 30 (2), 175-180, 1981.

J. Onishi, S. Kimura, R. J. W. James and Y. Nakagawa, "Solving the redundancy allocation problem with a mix of components using the improved surrogate constraint method", *IEEE Transactions on Reliability*, 56 (1), 94-101, 2007.

J. E. Ramirez-Marquez, D. W. Coit and A. Konak, "Redundancy Allocation for Series-Parallel Systems Using a Max-Min Approach", *IIE Transactions*, 42 (21), 891-898, 2004.

P.-S. You and T.-C. Chen, "An efficient heuristic for series-parallel redundant reliability problems", *Computer and Operations Research*, 32 (8), 2117-2127, 2005.

J-H. Zhao, Z. Liu and T.-M. Dao, "Reliability optimization using multiobjective ant colony system approaches", *Reliability Engineering and System Safety*, 92 (1), 109-120, 2007.

L. Zia and D.W. Coit, "Redundancy allocation for series-parallel systems using a column generation approach", Working Paper 05-027, Industrial and Systems Engineering, Rutgers University, 26 p., 2005.