Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)

Determination of Pruned Pareto Sets for the Multi-Objective System Redundancy Allocation Problem

Sadan Kulturel-Konak and David W. Coit

***Abstract*—In this paper, a new methodology is presented to solve multi-objective system redundancy allocation problems. A tabu search meta-heuristic approach is used to initially find the entire Pareto set, and then a Monte-Carlo simulation provides a decision maker with a pruned set of Pareto solutions based on decision maker's predefined objective function preferences. We are aiming to create a bridge between Pareto optimality and single solution approaches.**

I. Introduction

In many actual engineering problems, the decision maker is faced with solving multi-objective problems to find at least one feasible solution that can be implemented in the system design. When multiple objectives exist, the problem is either solved for a single solution or for an entire Pareto optimal set.

A Pareto optimal set is a set of solutions that are all non-dominated with respect to each other. While moving from one Pareto-optimal point to another, there is always a certain amount of sacrifice in one objective to achieve a certain advantage in the other. For any two solutions (decision vectors) $\mathbf{a}$, $\mathbf{b} \in \mathbf{X}$ ($\mathbf{X}$ is the feasible region) for a maximization problem, $\mathbf{a}$ dominates $\mathbf{b}$, or $\mathbf{b}$ is inferior to $\mathbf{a}$ if $f_i(\mathbf{a}) \geq f_i(\mathbf{b}) \; \forall i \in \{1,2,...,K\}$ and $\exists j \in \{1,2,....,n\}$ where $f_j(\mathbf{a}) > f_j(\mathbf{b})$. All solutions which are not inferior with respect to another solution form a set called the Pareto optimal set.

In this paper, we consider decision maker's preferences about the objectives which can be done in three ways [1]. The first way is prior to solving the problem: known as *a priori* method. This can be done by combining the multiple objectives into one composite function, which is then solved for an optimal solution. This single objective can be formulated and solved using methods such as utility theory [2] or value function [3] or weighted sum [4]. In all of these methods, however, choosing the right utilities or weights is a challenge, and different settings can result in very different final solutions. There are other methods as well including goal programming [5] and the ε-constraint method [6]. The second way is to specify the decision maker's preferences during the search process of the algorithm. Such methods are known as *interactive* where the decision maker guides the search to desirable parts of the solution space ([7]-[8]). The third way is the *posterior* method where a list of Pareto optimal solutions is obtained first, and then, the decision maker preferences are applied to find solution(s). This method can be preferable if the decision maker wants to consider different preferences, and compare solutions by giving different priorities to the objectives in order to observe how the solutions change. A summary of these kinds of approaches based on evolutionary optimization are provided by Coello Coello [9] and Branke and Deb [10].

There are also various meta-heuristics available to achieve Pareto optimal solution sets. Different forms of Tabu Search (TS) have been applied to multi-objective problems, such as Multi-objective TS (MOTS) by Hertz *et al*. [11], the modified version of the MOTS algorithm by Baykasoglu *et al.* [12], TS for Multi-objective Combinatorial Optimization (TAMOCO) by Hansen [13], and Multinomial Tabu Search (MTS) by Kulturel-Konak *et al*. [14]. Vector Evaluated Genetic Algorithm (VEGA), developed by Schaffer [15], is the first algorithm designed to cope with multiple objectives simultaneously by exploiting the parallel properties of Genetic Algorithms (GAs). Then, the Multi-objective GA (MOGA) is introduced by Fonseca & Fleming [16] and Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler & Thiele [17]. Multi Objective Simulated Annealing (MOSA), developed by Ulungu *et al*. [18], determines an efficient approximation to the Pareto frontier for the resolution of multiple objective combinatorial problems.

For many problems, the Pareto optimal set may include hundreds or thousands of solutions. Eventually, the decision maker has to analyze a large number of nondominated solutions and select the best one among to implement, which can be a very challenging task.

To alleviate this problem, several new methods, such as smart Pareto set by Mattson *et al*. [19], a relative objective weighting scheme by Kasprzak & Lewis [20], have been developed that reduce the Pareto optimal set to a set of solutions that is attractive to the decision maker and are pragmatically sized to help the decision making process.

Manuscript received January 2, 2007.

S. Kulturel-Konak is with Management Information Systems, Pennsylvania State University Berks, Reading, PA 19610, USA (corresponding author, phone: 610-396-6310; fax: 610-396-6024; e-mail: sadan@psu.edu).

D. W. Coit is with Department of Industrial & Systems Engineering , Rutgers University, Piscataway, NJ 08844, USA (e-mail: coit@rutgers.edu).

Methods solving a multi-objective problem similar to a single objective problem return a single solution. If a decision-makers can accurately assign meaningful weights or select appropriate value or utility functions, then that is the preferred approach. However, these methods can be problematic for many practitioners, because assigning appropriate numerical values (i.e., weights) to an objective function can be challenging and/or confusing. Therefore, the purpose of our study in this paper is to create a bridge between Pareto optimality and single solution approaches. The approach is specifically for the case when decision-makers can prioritize or ordinally rank the importance of the different objectives, but can not reliably assign numerical weights to combine the different objectives into a single objective function.

The first phase of the proposed approach generates a Pareto optimal set which is filtered in the second phase to obtain a set of solutions that are much smaller in size and attractive to the decision maker. The method is effective when especially the decision maker wants to consider different preferences. Once the whole set of Pareto front was obtained in the first step of the algorithm, the decision maker may change his/her preference, and run the second step with different preferences.

II. Methodology

A new method is presented to solve the multi-objective system design problems, e.g., system Redundancy Allocation Problem (RAP). A more extended version of this method, which also considers the uncertainty in reliability estimates, is presented in [21]. For multi-objective system RAP, the objectives include maximizing system reliability, minimizing system weight, minimizing system cost, and possibly other objectives. The proposed approach does not require explicit objective function numerical weights or utility functions, but it provides prioritized Pareto optimal solutions based on a non-numerical ranking of the importance of scaled objectives.

Our proposed methodology includes two steps. First, a TS meta-heuristic approach is used to initially find the Pareto-optimal front to the multi-objective optimization problems. Second, prioritized Pareto-optimal solutions are found by the Pareto Front Pruning (PFP) method which is based on a non-numerical ranking of the importance of scaled objectives. Monte-Carlo simulation of randomly selected and prioritized objective weights is used to identify promising solutions. Randomly selected weights are generated from an uncertain weight function, $f_{\mathbf{w}}(\mathbf{w})$, that characterizes the likelihood of possible weight combinations based on the decision makers objective function rankings.

A. Redundancy Allocation Problem

The objective of the RAP is most often to determine an optimal system design to maximize system reliability given constraints on the system. The RAP has been traditionally and extensively studied as a single objective problem in the literature (see [22]). In addition, a multi-objective version of the problem is also realistic and has been widely studied because not only it is applicable and relevant, but also, it is challenging to solve.

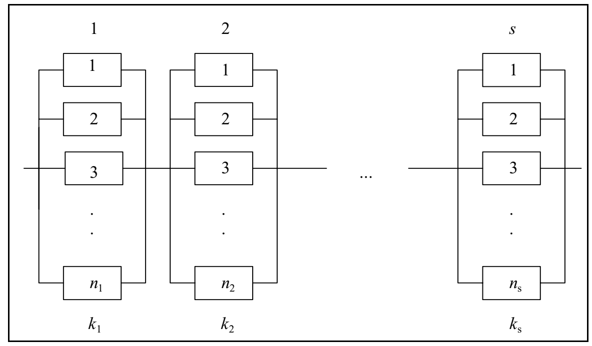


Fig. 1. Series-Parallel System Configuration

An example series-parallel system can be seen in Fig. 1. For each subsystem, there are multiple, functionally equivalent component that are available to be used in the system. The design can include a single component selection for each subsystem, or there may be multiple components selected and arranged in parallel. The decision variables are the component choices and the redundancy levels. Kuo & Prasad [23] provides a comprehensive review on system reliability optimization. For series-parallel systems with a single objective, Chern [24] demonstrated that the RAP is NP-hard.

Fyffe et al. [25] used dynamic programming to solve the RAP by limiting the problem to only one type of component available for each subsystem. Nakagawa & Miyazaki [26] demonstrated that a surrogate constraints approach is efficient to accommodate multiple constraints with dynamic programming. Integer programming [27], heuristics based on integer programming [28]-[29], GAs [30], and TS [22] have also been applied to determine the optimal design configuration.

In this paper, we are trying to find the optimal design configuration to maximize system reliability (R), to minimize system cost (C), and to minimize weight (W) when there are multiple component choices available for each of several k-out-of-n:G subsystems. Therefore, the mathematical formulation of the problem is given below.

$$\left.\begin{array}{l} \max\ R = \prod_{i=1}^{s} R_i(\mathbf{x}_i \mid k_i), \\ \min\ C = \sum_{i=1}^{s} C_i(\mathbf{x}_i), \\ \min\ W = \sum_{i=1}^{s} W_i(\mathbf{x}_i) \\ \text{subject to} \\ \qquad k_i \le \sum_{j=1}^{m_i} x_{ij} \le n_{\max,i} \qquad \forall i = 1, 2, ..., s \end{array}\right\} \qquad (1)$$

Notation:

R, C, W	reliability, cost, and weight of the system, respectively
s	number of subsystems
$\mathbf{x}_i$	$(x_{i1}, x_{i2}, ..., x_{i,m_i})$
x_{ij}	quantity of j^{th} component in subsystem i
n_i	total number of components used in subsystem i (i.e., $\sum_{j=1}^{m_i} x_{ij}$)
$n_{\max,i}$	user assigned maximum number of components in parallel used in subsystem i
k_i	minimum number of components in parallel required for subsystem i
$R_i(\mathbf{x}_i/k_i)$	reliability of subsystem i, given k_i
$C_i(\mathbf{x}_i)$	total cost of subsystem i
$W_i(\mathbf{x}_i)$	total weight of subsystem i
P	Final set of nondominated solutions
r_i	rank of objective i
K	number of objectives
T	number of simulation replications

I

B. The Multinomial Tabu Search Algorithm

The multinomial tabu search (MTS) algorithm developed by Kulturel-Konak et al. [14] is used to generate a Pareto front. In practice, any meta-heuristic approach could be used instead of the MTS algorithm, since the main research contribution in this paper is pruning the Pareto- optimal front. MTS algorithm [14] is a canonical TS procedure with a simple diversification which directs the search starting from a random point whenever nondominated solution list has not been updated for a predefined number of moves. Further information about canonical and more complex versions of TS may be seen in [31] and [32].

C. The Pareto Front Pruning Method

The pre-step of the pruning method is obtaining a full Pareto optimal set using MTS. The objective functions are then normalized and prioritized (ranked) according to the decision maker's needs. The decision maker ranks the objectives as the most important, second most important, etc. (ties are allowed). Random objective function weight assignments (w_i) are then repetitively selected from a joint probability density function capturing the previously determined sequence of ranked objectives. In each iteration, the multi-objective function is then transformed into a single objective function, and the "best" solution obtained is recorded for that particular weight combination. This procedure is repeated numerous times resulting in a group of solutions that appear as the "best" solution most often. This approach can dramatically reduce the size of the Pareto solutions to be presented to the decision maker. These pruned Pareto solutions can then be categorized into priority groups, based on the selection frequency, as an aid to the decision maker.

<u>The PFP Algorithm for the maximization of all objectives:</u>

$c(\mathrm{x}) \leftarrow 0$ for $\mathbf{x} \in P$

Normalize each objective $f_i(\mathbf{x}) = f_i(\mathbf{x})/(f_i^{\max} - f_i^{\min})$ for $\mathbf{x} \in P, i = 1,..,K$

Rank the objectives

For t=1...T Do {

Based on the ranks, generate a random weight vector $\{w_1, w_2, \ldots, w_K\}$ such that $w_i \le w_j$ if $r_i \le r_j$

Normalize the weights such that $\sum_{i=1}^{K} w_i = 1$

Calculate $f'(\mathbf{x}) = \sum_{i=1}^{K} w_i f_i(\mathbf{x})$ for $\mathbf{x} \in P$

Find the best solution $\mathbf{x}^*$ such that $f'(\mathbf{x}^*) = \max_{\mathbf{x} \in P}\{f'(\mathbf{x})\}$

$c(\mathbf{x}^*) \leftarrow c(\mathbf{x}^*)+1$}

Remove each solution $\mathbf{x}$ from P if $c(\mathbf{x})=0$

Group the solutions based on $c(\mathbf{x})$

The new method requires the decision maker to sort (or rank) the objective functions in order of their priority. There are different ways to rank the objective functions as well as a great deal of flexibility allowed in this method. For example, one objective can be selected as the most important objective and assigned a rank of one and a rank of two can be assigned to the rest of the objectives. This ranking scheme is different from assigning pre-selected weights, or utility functions. An example of ranking objective functions:

<u>Preference:</u> Objective $f_1(\mathbf{x})$ is more important than objective $f_3(\mathbf{x})$, Objective $f_3(\mathbf{x})$ is more important than objective $f_2(\mathbf{x})$

Ranked objectives = $\{f_1(\mathbf{x}), f_3(\mathbf{x}), f_2(\mathbf{x})\}$: $r_1 > r_3 > r_2$

The next stage is to generate a random weight function $\{w_1, w_2, \mathrm{w}_3\}$ such that $w_1 > w_3 > w_2$. An uncertain weight function is derived and used to generate random but ranked weight sets that can be used to filter solutions to form a realistically sized solution set. This set of solutions represents the decision maker's preferences in the objective functions, and the size of the set enables him or her to analyze and identify the solution of his or her choice with ease. From the original Pareto set, there will be a "best" solution for a particular weight combination. This best solution is noted and recorded, and then, the process is continued numerous times.

III. Numerical Examples

The two-step approach (the combination of MTS and PFP) is applied successfully to an example series-parallel system RAP. The test problem considered here uses the input parameters of the problem originally proposed by Fyffe *et al*. [25]. The component cost, weight, and reliability values are given in [25]. An unconstrained version of the problem is studied with a pre-defined preference of the decision maker. All algorithms are coded in C and run using an Intel Pentium IV with 2.6 GHz processor and 1.5 GB RAM.

Initially, MTS was applied to determine a Pareto optimal set. A permutation encoding of size $\sum_{i=1}^{s} n_{\max,i}$, representing a concatenation of the components in each subsystem including non-used components (i.e., defined as "blanks" when $n_i < n_{\max,i}$) is used. To obtain an initial feasible solution, for each subsystem, s integers between k_i and $n_{\max,i} - 3$ (inclusive) are chosen according to a discrete uniform distribution to represent the number of components in parallel (n_i). Then, the n_i components are randomly and uniformly assigned to the m_i different types. If feasible, it becomes the initial solution. Then, one of the problem objectives is equally and randomly chosen as an active objective to evaluate candidate solutions. Moves operate on each subsystem. The first type of move changes the number of a particular component type by adding or subtracting one. These moves are considered individually for all available component types within all subsystems. The second type of move simultaneously adds one component to a certain subsystem and deletes another component, of a different type, within the same subsystem. The two types of moves are performed independently on the current solution. The structure of the subsystem that has been changed in the accepted move is stored in the tabu list. The dynamic tabu list length is changed according to a uniform integer random number between [s, $3s$]. Finally, the stopping criterion is defined as the maximum number of iterations conducted without updating the ND solutions list and set to 1,000. Furthermore, for this problem, the decision maker's *preference of the objectives* is in the following order: Maximizing R, Minimizing C, and Minimizing W (i.e., the most important objective is Maximizing R, then Minimizing C is the second most important, and Minimizing W is the least important). It is a minor change in the algorithm to adopt different preferences (see [21]).

A. *Unconstrained Three Objective Problem*

We solved the unconstrained version of the problem (1), and we have three objectives of maximizing R, minimizing C and W. The above stated preference was applied to prune the Pareto optimal set found using MTS, and 10,000 randomly selected weights were obtained to develop the pruned Pareto set. The pruned Pareto set is then subjectively divided into priority groups based on selection frequency. The priority groups provide an indication of which solutions have similar number of selections. If a particular solution is selected more often, it represents a larger region of possible weight combinations given the objective function preferences.

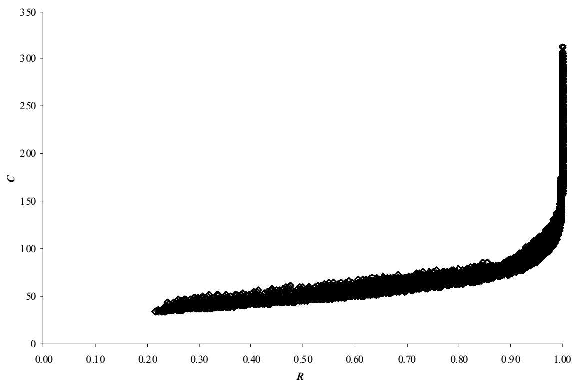
Fig. 2 shows the comparison of the Pareto sets before and after pruning with the pre-defined preference. As seen in the figure, the pruning process significantly decreases the number of solutions in the Pareto front. As opposed to 8,035 Pareto optimal solutions found using MTS, pruned Pareto front includes only 80 solutions. The benefit of the new pruning technique can be further seen in Table I which displays the frequency and cumulative percentage of the pruned Pareto solutions. In this table, the pruned Pareto optimal solutions are further divided into priority groups for decision makers. By following the Pareto principle, we can see that only a few solutions are identified as prioritized solutions. (i.e., four solutions are encountered 41.84% of time with the decision maker's preference.)

IV. Conclusion

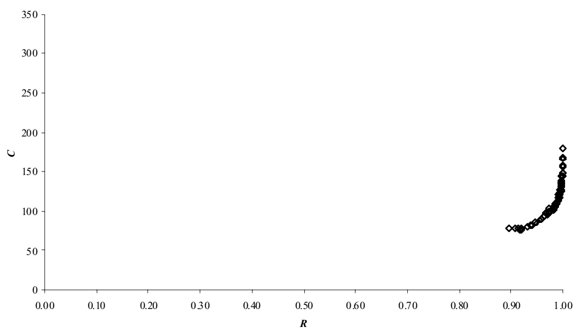
The proposed approach is to reduce the size of the final set of non-dominated solutions found by a meta-heuristic approach. From a decision maker perspective, it is usually more intuitive and easier to prioritize the objectives than to assign them individual weights. Therefore, the proposed method helps achieve a balance between a Pareto optimal solution set and a single solution, so that the Pareto set is pruned. The pruned Pareto set gives the decision-maker a reasonable sized set of solutions that match his/her preferences.

TABLE I
PARETO PRUNED SOLUTIONS IN PRIORITY GROUPS

Group	# of Solutions	Total Counter	Frequency	%	Cum. %
1	4	4284	0.42	41.84	41.84
2	5	2633	0.26	26.33	68.17
3	13	2207	0.22	22.07	90.24
4	24	976	0.10	9.76	100.00



(a) Pareto set found by MTS



(b) Pruned Pareto set with the pre-defined preference

Fig. 2. Pareto Sets Before and After Pruning

REFERENCES

[1] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: analyzing the state-of-the-art," *Evolutionary Computation*, vol. 8, pp. 125-47, 2000.

[2] P. C. Fishburn, *The Foundations of Expected Utility (Theory and Decision Library)*. Boston: Kluwer Academic Publishers, 1982.

[3] R. Goebel, "Hamiltonian dynamical systems for convex problems of optimal control: implications for the value function," Las Vegas, NV, USA, 2002.

[4] A. M. Geoffrion, "Proper Efficiency and the Theory of Vector Maximization," *Journal of Mathematical Analysis and Application*, vol. 22, pp. 618-630, 1968.

[5] A. Charnes and W. W. Cooper, *Management Models and Industrial Applications of Linear Programming*. New York: John Wiley & Sons, 1961.

[6] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*: John Wiley & Sons, 2001.

[7] H. Yano, "Interactive Fuzzy Decision Making for Multiple Decision Maker-Multiple Objective Programming Problems with Fuzzy Parameters," presented at 17th International conference on Multiple Criteria Decision Making, 2004.

[8] N. Palli, S. Azram, P. Mccluskey, and R. Sundarajan, "An Interactive Multistage Epsilon- Inequality Constraint Method for Multiple Objective Decision Making," *Journal of Mechanical Design*, vol. 120, pp. 678-686, 1998.

[9] C. A. Coello Coello, "Handling Preferences in Evolutionary Multi-objective Optimization: A Survey," presented at Congress on Evolutionary Computation, 2000.

[10] J. Branke and K. Deb, "Integrating User Preferences into Evolutionary Multi-objective Optimization," in *Knowledge Incorporation in Evolutionary Computation*, Y. In Jin, Ed.: Springer, 2004, pp. 461-477.

[11] A. Hertz, B. Jaumard, C. C. Ibeiro, and W. P. Formosinho Filho, "A Multi-Criteria Tabu Search Approach to Cell Formation Problems in Group Technology with Multiple Objectives," *Recherche Operationelle/Operations Research*, vol. 28, pp. 303-328, 1994.

[12] A. Baykasoglu, S. Owen, and N. Gindy, "A Taboo Search Based Approach to Find the Pareto Optimal Set in Multiple Objective Optimisation," *Journal of Engineering Optimization*, vol. 31, pp. 731-748, 1999.

[13] M. P. Hansen, "Tabu search for multiobjective combinatorial optimization: TAMOCO," *Control and Cybernetics*, vol. 29, pp. 799-818, 2000.

[14] S. Kulturel-Konak, A. E. Smith, and B. A. Norman, "Multi-objective tabu search using a multinomial probability mass function," *European Journal of Operational Research*, vol. 169, pp. 918-931, 2006.

[15] J. D. Schaffer, "Multiple Objective optimization with vector evaluated genetic algorithms," presented at International Conference on Genetic Algorithm and their applications, 1985.

[16] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization," presented at Proceedings of ICGA-93: Fifth International Conference on Genetic Algorithms, 17-22 July 1993, Urbana-Champaign, IL, USA, 1993.

[17] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 257-71, 1999.

[18] B. Ulungu, J. Teghem, and C. Ost, "Efficiency of Interactive Multi-Objective Simulated Annealing through a Case Study," *Journal of the Operational Research Society*, vol. 49, pp. 1044-1050, 1998.

[19] C. A. Mattson, A. A. Mullur, and A. Messac, "Smart Pareto Filter: Obtaining a Minimal Representation of Multiobjective Design Space," *Engineering Optimization*, vol. 36, pp. 721-740, 2004.

[20] E. M. Kasprzak and K. E. Lewis, "Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method," *Structural and Multidisciplinary Optimization*, vol. 22, pp. 208-218, 2001.

[21] S. Kulturel-Konak, D. W. Coit, and F. Baheranwala, "Pruned Pareto Optimal Sets for the Systems Redundancy Allocation Problem Based on Multiple Prioritized Objectives," *Journal of Heuristics*, submitted for publication, 2006.

[22] S. Kulturel-Konak, A. E. Smith, and D. W. Coit, "Efficiently solving the redundancy allocation problem using tabu search," *IIE Transactions*, vol. 35, pp. 515-26, 2003.

[23] W. Kuo and V. Prasad, "An Annotated Overview of System-Reliability Optimization," *IEEE Transactions on Reliability*, vol. 49, pp. 487-493, 2000.

[24] M. S. Chern, "On the computational complexity of reliability redundancy allocation in a series system," *Operations Research Letters*, vol. 11, pp. 309, 1992.

[25] D. E. Fyffe, W. W. Hines, and N. K. Lee, "System Reliability Allocation and a Computational Algorithm," *IEEE Transactions on Reliability*, vol. 17, pp. 74-79, 1968.

[26] Y. Nakagawa and S. Miyazaki, "Surrogate Constraints Algorithm for Reliability Optimization Problems with Two Constraints," *IEEE Transactions on Reliability*, vol. R-30, pp. 175-180, 1981.

[27] M. Ghare and R. E. Taylor, "Redundancy for reliability in series system," *Operations Research Letters*, vol. 17, pp. 838-847, 1969.

[28] J. E. Ramirez-Marquez, D. W. Coit, and A. Konak, "Redundancy Allocation for Series-Parallel Systems Using a Max-Min Approach," *IIE Transactions*, vol. 41, pp. 891-8, 2004.

[29] D. W. Coit and A. Konak, "Multiple weighted objectives heuristic for the redundancy allocation problem," *IEEE Transactions on Reliability*, vol. 55, pp. 551-8, 2006.

[30] D. W. Coit and A. E. Smith, "Reliability optimization of series-parallel systems using a genetic algorithm," *IEEE Transactions on Reliability*, vol. 45, pp. 254-60, 1996.

[31] F. Glover, E. Taillard, and D. de Werra, "A user's guide to tabu search," *Annals of Operations Research*, vol. 41, pp. 3-28, 1993.

[32] F. Glover and M. Laguna, *Tabu Search*. Boston: Kluwer Academic Publishers, 1997.