AVAILABILITY COST OPTIMIZATION METHODOLOGY DESCRIPTION
STEVE HALL
ASSET OPTIMIZATION SERVICES

Life Cycle Cost Optimization Process[1]

Availability and efficiency improvements at power plants can be achieved at the unit level by selectively improving component reliability, maintainability, or efficiency. However, implementation of a specific availability or efficiency improvement or series of improvements may not prove to be cost effective because the gain expected from implementing the improvement is less than the cost of the improvement or because of the imposition of outside constraints such as scheduled outage time and funding limits. The objective of the life cycle cost optimization process is to select those component improvements that will provide an increase in availability or efficiency and also reflect the greatest net benefit within imposed resource constraints (funding, schedule, manpower).

The improvement life cycle cost optimization process employs a four-step iterative approach as illustrated in Figure 1. The first step is to collect the information and data related to the improvements under evaluation. The second step is to apply an economic screening criteria and method to determine which improvement options are potentially cost beneficial. The third step considers various constraints such as funding limitations, outage schedules, and manpower limitations to further evaluate the candidate improvements. The final step of the approach is to evaluate surviving candidate improvements through a dynamic program algorithm to arrive at a sequence of improvements that provide the greatest net benefit within established constraints.

DATA COLLECTION

In order to implement the life cycle cost optimization process it is necessary to establish a relationship between the cost of implementing an improvement and the expected benefit of that improvement. That relationship is established by determining the cost of the improvement, estimating the expected increase in component availability resulting from that improvement, calculating the effect of the component availability change on overall unit equivalent availability or capacity factor and converting the change in unit equivalent availability into a benefit based on an increase in net generation revenue. To accomplish that, the following information is required:

- A listing of the reliability, availability, maintainability (RAM), and efficiency improvement options under consideration
- The cost required to implement each improvement option
- The time required to implement each change
- For RAM improvements, the actual or estimated change in event frequency and/or downtime resulting from each improvement option
- For efficiency improvements, the expected percent increase in net revenue from either decreasing the fuel cost or in increasing net generation capability.
- An LCC simulation model and associated baseline data for the plant (or plants) to be evaluated
- The cost relationships between unit availability and costs such as replacement power, fuel, and operations and maintenance expenditures

[1] The information in this paper is based on work the author undertook in relation to developing a solution for EPRI in 1989 that used the UNIRAM RAM modeling methodology. The processes described herein are in the public domain.

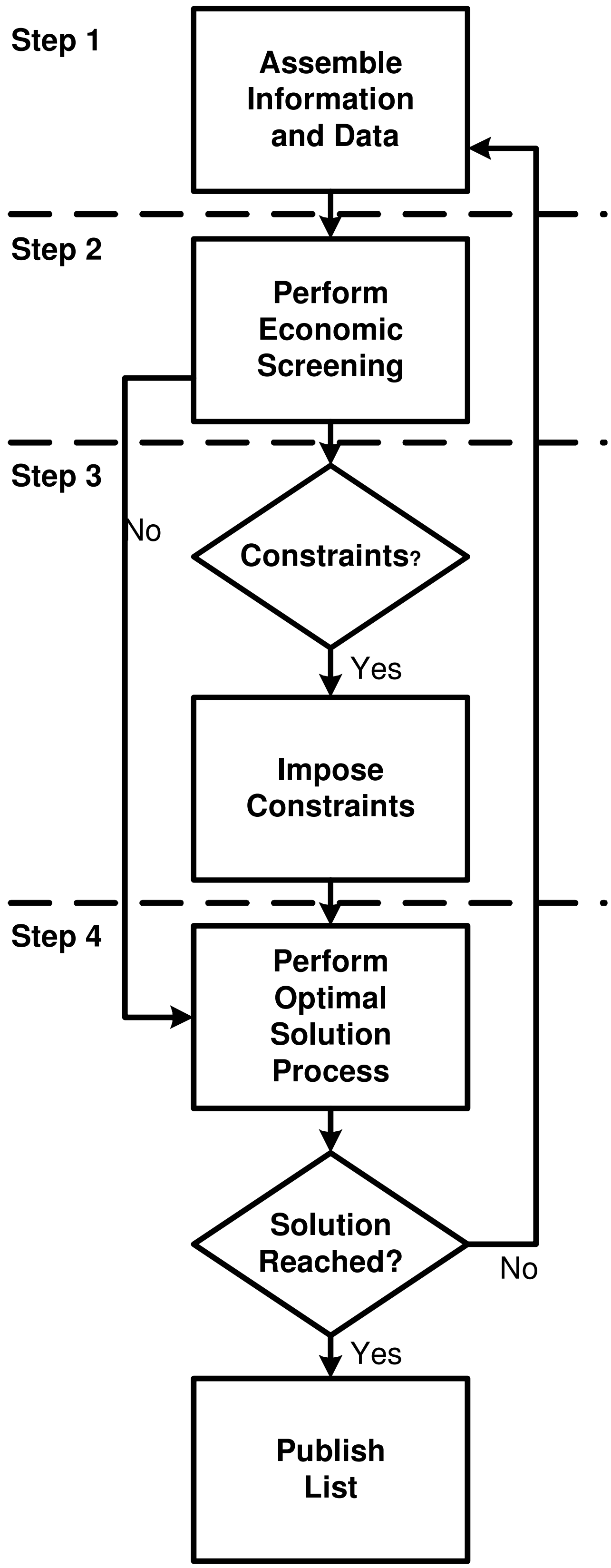

Figure 1. Availability Optimization Process

- Identification of funding, schedule, or other resource constraints
- Economic factors such as escalation, discount, and interest rates
- Unit production demand parameters (e.g., baseload, cycling, peaking)

The LCC simulation model is used to assess changes in unit availability that may occur due to changes in component RAM characteristics so that the relationship between availability and production costs can be studied quickly and accurately. The need for information relating to constraints is required because the cost optimization methodology must be responsive to the possibility of limited capital, outage time, or the labor and engineering resources available for implementing improvements. This is especially true for improvement projects that must compete for funding.

ECONOMIC SCREENING ANALYSIS

An economic screening analysis is used to identify those candidate improvement options that have the potential for producing a positive net benefit. This initial economic screening assumes that the proposed improvements are independent. Before beginning this analysis, a LCC simulation is performed for the plant (or plants) to evaluate the effect that changes in component availability have on unit production. The output of the LCC simulation is a criticality ranking (C_i). This ranking indicates, for each component or event, the increase in unit productivity to be expected if that component were to achieve "perfect" availability, i.e., its availability becomes 1. The forecasted change in component availability for a given proposed improvement (ΔA_c) is then multiplied by he component's criticality ranking (C_i x ΔA_c) to calculate the approximate change in unit availability that can be expected from implementing that change. This initial screening relies on the assumption that the relation ship between component and unit availability is linear. As Figure 2 illustrates, this relationship is typically non-linear. However, the relationship can be linearly approximated for small changes in component availability. For each proposed RAM improvement, the expected increase in unit production is then used to estimate the increase in annual megawatt hours that may be expected from a specific component improvement. To calculate the change in expected megawatt hours (ΔMW-HR), the following equation is used:

$$\Delta\text{MW-HR} = \Delta A_u \times (\text{Unit Net Capacity}) \times (\text{Scheduled Operating Hours}) \quad (1)$$

The increase in power production can then be converted to an expected revenue increase and compared to the cost of making the component improvement.

- Improvements proposed for increasing efficiency would be economically screened as follows:

  If the proposed change results in an increase in net capacity, the benefit would be calculated using the following equation:

  $$\Delta\text{MW-HR} = A_u \times \Delta(\text{Unit Net Capacity}) \times (\text{Scheduled Operating Hours}) \quad (2)$$

  As before, the increase in power production can then be converted to an expected revenue increase compared to the cost of making the component improvement.

- If the proposed change resulted in lowering fuel costs, i.e., less fuel is required to generate the same amount of power the following equation would be used to calculate the expected benefit:

  $$\text{Benefit} = \text{MW-HR's} \times \Delta(\text{Cost/MW-HR}) \quad (3)$$
  Where $\text{MW-HR's} = A_u \times (\text{Unit Capacity}) \times (\text{Scheduled Operating Hours})$

Should there be other cost factors affected by changes in unit productivity, these too can be estimated in a similar manner. Those component improvements that would provide a cost savings greater than the investment cost then become *potential* economically viable improvement candidates because, as we will see later, they maybe dropped from consideration for other reasons. If so desired, the present worth of the costs and benefits can be used in the economic screening process to account for the time value of money over the life of the change.

It is recognized that depreciation and tax effects could also be added to the economic screening model, as can O&M costs. Although these are second order effects they can be included should these factors be significant.

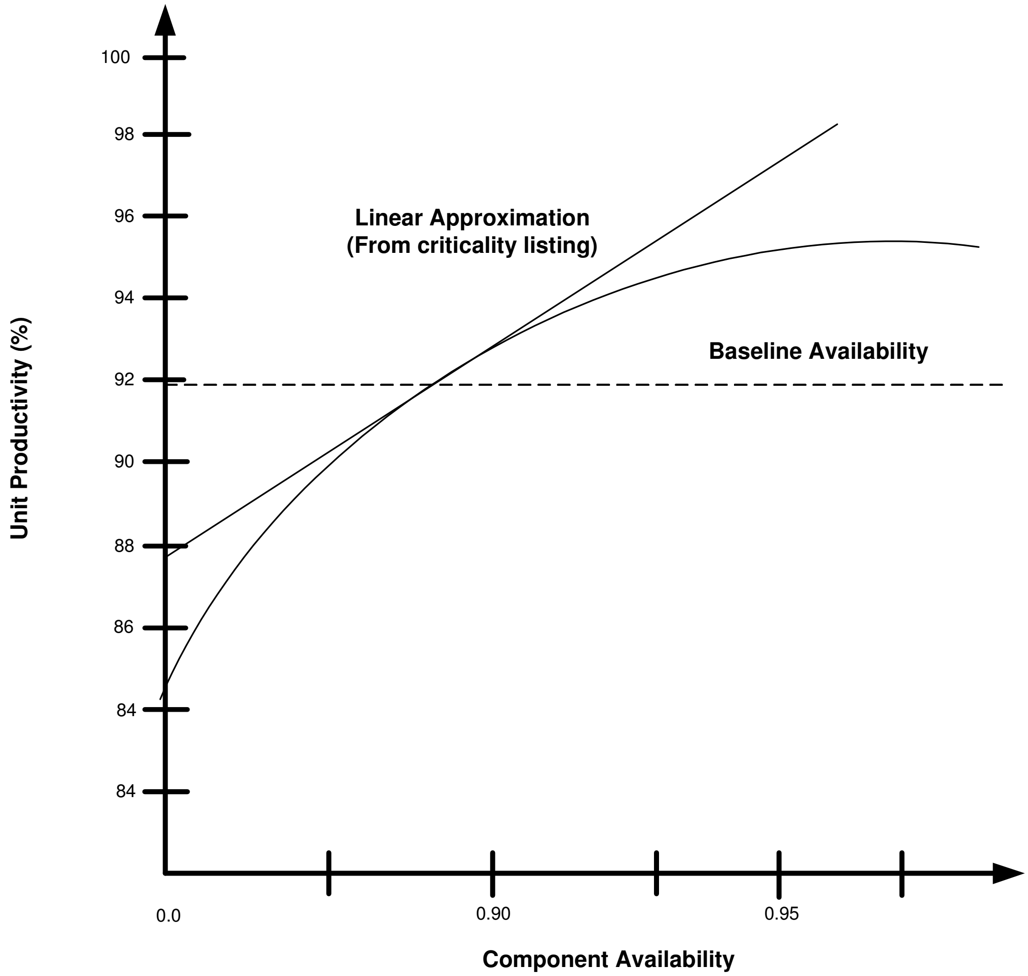

Figure 2. Component/Unit Availability Relationship

The output of the economic screening process is a list of potential economically viable improvement projects. These projects, with their costs and benefits, are then analyzed considering additional constraints (e.g., minimum cost-benefit ratio, must do for regulatory reasons, negative impact on safety, etc.) that may be desired.

OPTIMIZATION WITH CONSTRAINT

The third step of the analysis considers any stated constraints on the improvement process such as funding limitations or manpower resources. If there are no constraints, or the constraints are not exceeded, the optimization process can proceed to the optimal solution process. If the limitations of any constraints are not satisfied, an integer program (IP) algorithm is applied to the economically screened candidate improvement options prior to last step. The objective of the IP algorithm step is to choose the combination of improvements that provide the optimum benefit while satisfying the limitations of each constraint. An example of the IP process is contained in Appendix A. The IP step assumes that the benefit resulting from each specific improvement will not affect the benefit of other improvements and that the total benefit is the sum of each individual benefit. .

The result of using the IP is a list of candidate component improvements that maximize the net benefits and meet the imposed constraints. If the assumption of independence and linearity reflected the actual relationship between component and unit availability, the IP would provide the final optimum set of improvements. However, as seen in Figure 2, the relationship between component availability and unit availability is often non-linear and experience with LCC simulation models has shown that component improvement effects are not independent.

OPTIMAL SOLUTION PROCESS

The final step in the optimization process is to apply a dynamic programming (DP) algorithm to the set of candidate improvement options. The objective of the DP algorithm is to optimize the solution set taking into account any non-linearities that exists between component and unit availability and any interdependency that can exist between components. This is done by making a sequence of selections such that if the process were prematurely terminated, the changes selected to that point would still be optimal.

As each component improvement is selected and the baseline design of the unit is changed (via the LCC model), it can be expected that the ratio of changes in unit availability to changes in component availability of the unmodified components will either increase, decrease, or remain the same. Because of these changes, it is possible that some component improvements that were previously not cost beneficial will become beneficial. Conversely, it is also possible that some improvements will no longer be beneficial. The unpredictable effect of changes on component criticalities (C_i) is investigated using the DP algorithm. An example of this algorithm is contained in Appendix B. Note that this DP algorithm is dependent on the same constraints imposed by the IP algorithm.

The DP algorithm is a process that methodically addresses the expected benefit of implementing alternative sets of improvement candidates to ascertain the set that will provide the greatest net benefit. As each improvement candidate is implemented and the baseline design of the unit is changed (via the LCC model), the economic screening and imposition of constraints processes are again done on an iterative basis. The economic screening is accomplished with the new baseline design; the imposition of constraints is accomplished with a reduction in the constraint equal to the cost of the candidate improvement(s) implemented.

The result of these optimal analyses is a chronologically ordered list of recommended improvements that should provide maximum return on an improvement investment considering all constraints.

Illustration of Integer Programming (IP) Algorithm[2]

In order to illustrate the application of the Integer Programming (IP) algorithm, it is assumed that a life cycle cost model (LCCM) has established a set of seven economically viable component improvement options and their associated estimated expected benefits (ΔB_i) and implementation (IC_i) cost values. These are summarized in Table A-1. It is further assumed there exists a cost constraint of $1,000,000. The problem is then:

Maximize: $$\sum_{i=1}^{7} \Delta B_i X_i \qquad \text{(A-1)}$$

Subject to: $$\sum_{i=1}^{7} IC_i X_i < 1{,}000{,}000 \qquad \text{(A-2)}$$

$$X_i = 0,1$$

Alternative	Estimated NPW Benefit (ΔB_i)	Implementation Cost (IC_i)
X_1	$157,000	$151,000
X_2	$640,000	$362,000
X_3	$207,000	$477,000
X_4	$273,000	$100,000
X_5	$95,900	$230,000
X_6	$17,100	$$37,000
X_7	$148,000	$959,000
		$2,316,900

Table A-1. Sample Problem Data

The first step in implementing the solution is to order the alternatives in order of decreasing benefit. For this example the ordered set is:

X_2, X_4, X_3, X_1, X_7, X_5, X_6

The operation of the algorithm for this example is summarized in Table A-2. The placing of a one (1) directly below a variable in the ordered set signifies the consideration of those variables for a potential solution. Each consideration is called an iteration and is one horizontal line in the algorithm solution. As an additional variable is considered in a subsequent iteration, it will be the next "leg" and will be indicated in the next vertical column. A leg begins what will be a path of possible combination of variables. The objective of the algorithm is to efficiently search the solution tree for the optimum solution, eliminating legs such that all possible combinations that will not result in an optimal solution will not have to be evaluated.

The constraint value (C_T) is the total cost of implementing the combination of variables of a given iteration. The maximum value (V) is the absolute best possible value if the indicated variable is implemented and assuming all the benefits from other legs that are not specified by a 1 or 0 are obtained. For example, in iteration 1 (in which variable X_2 is considered as a potential solution) the calculation assumes the benefits from all the alternatives in addition to X_2, thereby considering the best possible outcome from selecting X_2 as the root of a leg. The sum of the benefits from all alternatives in the sample problem data is $158,000, which is entered into the minimum value column.

[2] Egon Balas, *An Additive Algorithm for Solving Linear Programs with Zero-One Variables*, Operations Research, 13(4): 517-546, July –August, 1965

Iteration No.	Variable $X_2X_4X_3X_1X_7X_5X_6$	Constraint Value (C_T)	Maximum Value (V)	Stored Solution (V_S)	Eliminated? Yes or No	Reason 1,2, or 3
1	1	$362,000	$1,538,000	0	No	
2	1 1	$462,000	$1,538,000	0	No	
3	1 1 1	$939,000	$1,538,000	0	No	
4	1 1 1 1	$1,090,000	$1,538,000	0	Yes	1
5	1 1 1 0 1	$1,898,000	$1,381,000	0	Yes	1
6	1 1 1 0 0 1	$1,169,000	$1,233,000	0	Yes	1
7	1 1 1 0 0 0 1	$976,900	$1,137,100	0	Yes	3
8*	1 1 0 1 0 1 1	$880,000	$1,183,000	$1,137,100	Yes	3
9	1 0 1	$569,000	$1,265,000	$1,183,000	No	
10	1 0 1 1	$720,900	$1,265,000	$1,183,000	No	
11	1 0 1 1 1	$1,679,000	$1,265,000	$1,183,000	Yes	1
12	1 0 1 1 0 1	$950,000	$1,170,000	$1,183,000	Yes	2
13	1 0 1 0 1	$1,798,000	$1,108,000	$1,183,000	Yes	1,2
14	1 0 0 1	$513.000	$1,158,000	$1,183,000	Yes	2
15	0 1	100,000	$898,000	$1,183,000	Yes	2

*Solution

Table A-2. Algorithm Solution

To begin the operation, an initial stored solution is assumed (0 in this case since all alternatives are required to be positive). The decision tree is then explored and, at each iteration, either a leg is eliminated or no decision is obtained. If a leg is eliminated it is removed from further consideration. A leg will be eliminated if any of the following conditions is met:

1. The cost total (C_T) with the next leg will exceed the maximum cost constraint.
2. The next leg and the maximum contribution of the remaining legs are not as beneficial as the last stored solution (V_S).
3. The next leg is the last leg and creates an improved solution (V). (Note: This improved solution then becomes the new stored solution (V_S).

In iteration 1, none of the criteria for eliminating a leg are met because the cost of implementing X_2 is $362,000 against a constraint of $1,000,000 and the stored solution is 0. Therefore no decision is obtained in iteration 1. However in iteration 4, in which the total cost ($1,090,000) exceeds the total cost constraint, elimination occurs because of condition 1. Therefore this leg root, in which the first four variables are considered implemented, will be eliminated from further investigation entirely. Consequently, the next iteration considers the first three and the fifth variables implemented at a total cost of $1,898,000 which also exceeds the cost constraint again results in the elimination of a root.

Iterations 6 and 7 continue trying the next possible combinations of variables without an optimal solution. The solution in iteration 7 specifies the last leg and has a maximum value that is more beneficial than the stored solution and is therefore eliminated under condition 3. This process continues until all leg roots have either been eliminated or accepted and the most beneficial solution that satisfies the cost constraint is selected. Iteration 8 is selected as the optimal solution; it provides the largest maximum value for the imposed constraint.

This example yields a solution that calls for implementing candidates X_1, X_2, X_4, X_5, and X_6 with a maximum total increase in production revenue of $1,830,000 and a capital investment of $880,900 (within the $1,000,000 cost constraint). In this example 128 (2^7) potential candidate selections existed. However, through application of the IP algorithm, only fifteen needed to be identified for examination and those fifteen were examined fairly rapidly because of the building block nature of the process.

Illustration of the Dynamic Programming (DP) Algorithm

In the example presented in Appendix A, an optimal linear solution was found from evaluating seven candidates for a total capital cost of $880,900 which corresponds to a total revenue increase of $1,183,000. Because the relationship between component and unit availability is usually non-linear, a means of addressing those relationships and their affect on achieving an optimum solution is required. The application of a DP algorithm is used to address this requirement. This appendix illustrates how non-linear relationships can affect an optimal solution and demonstrates the use of the dynamic programming (DP) algorithm to address non-linear relationships that could impact the actual optimal solution.

Non-linear Relationship Effect

Figure B-1 can be used to illustrate how non-linear relationships can affect development of an optimal solution. Figure B-1 shows the relationship between component A and the unit and component B and the unit. Before any improvements are proposed, the relationships between the two components and the unit are depicted by the two curves and unit availability labeled as baseline. If the availability of only component A is improved as indicated by the arrow labeled A, unit availability would increase to about 93% (dotted line), a change of 1%. If the availability of only component B is improved as indicated by the arrow labeled B, unit availability would increase to about 95% (dashed line), a change of 3%. If, however, component A and then component B are implemented in sequence, the following occurs:

- Component A is improved and baseline availability moves to 93%
- Since there is yet no improvement in B, the curve representing the relationship between component B and the unit will shift as shown and the unit availability corresponding to the unchanged component B availability moves from B_1 to B_2. This occurs because the point at which the curve crosses the Y-axis does not change appreciably and the baseline availability of for component B does not change.
- Component B is now improved. However, unit availability increases by 4% to 97%.

Assume that the worth of improving availability by 1% is $1,000,000 ($20/MW-HR, 1000 MW unit). When the two components are considered independently, the benefit is equivalent to $4,000,000; when combined, the value is $5,000,000. While not illustrated in Figure B-1, if component B is implemented and then component A, the increase is only about 3.8%.

DP Algorithm

The first step in the DP algorithm is to create a matrix such as the one shown in Table B-1. Each column in the matrix corresponds to linear X_k solution variable (X_1, X_2, X_4, X_5, X_6 in the previous IP solution). IC_k^* on each column corresponds to C_T-IC_k (i.e., the cost constraint minus the implementation cost associated with the K^{th} candidate variable). V_k^* corresponds to the IP solution of the reduced problem (i.e., excluding X_k) plus the actual improvement expected from implementing X_k using the LCC simulation model. Each column in the matrix corresponds to a reduced problem solution given that X_k is selected as an improvement. Variables included in the reduced problem solution are identified by ones and by zeros if they are not. A zero is entered for X_k since it cannot enter the reduced solution as an improvement. Variables included in the reduced problem solution are identified by ones and by zeros if they are not. A zero is entered for X_k since it cannot enter the reduced solution.

Each reduced problem is solved using the following steps:

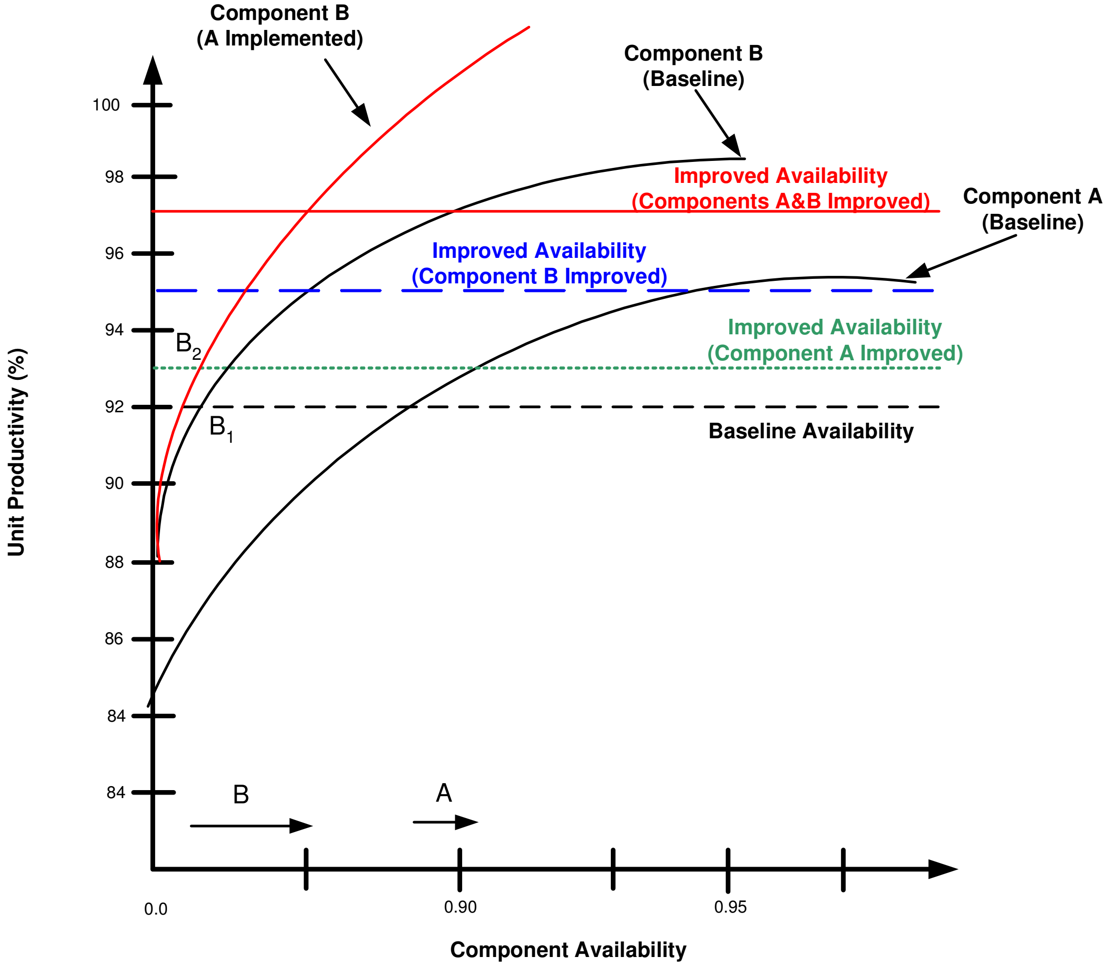

Figure B-1. Component/Unit Availability Non-linear Relationships

V_k^*					
IC_k^*	\$848,000	\$638,000	\$900,000	\$770,000	\$962,100
X_k	X_1	X_2	X_4	X_5	X_6
X_1	0				
X_2		0			
X_3					
X_4			0		
X_5				0	
X_6					0
X_7					

Table B-1. Initial DP Matrix

1. Candidate X_k is assumed to be implemented and the LCC simulation is executed to determine the expected change in MW-HR's associated with its implementation and to determine the criticality ranking of the remaining candidates. If the set of criticality ranking values are unchanged from the original baseline ranking values, the reduced solution will be the same as the original IP solution and the solution can proceed to the next X_k and its reduced problem.
2. A new set of ΔC_k's (where C_k is the benefit associated with the K^{th} improvement) are determined by converting ΔMW-HR's to Δ\$'s. At this point, because of non-linear effects, some of the variable candidates may drop out of the solution be cause the no longer supply a $\Delta C_k > 0$ (or other selected minimum value > 0). In this case zeros would be entered into the reduced

problem solution for the appropriate X_k's.

3. The IP algorithm is applied to the reduced problem, using IC_k^* as the constraint and the solution is entered into the appropriate column.
4. The value of V_k^* corresponding to the reduced problem solution value plus the value of implementation of the k[th] candidate is determined.

After all the reduced problems have been solved, the column having the maximum V_k^* value is selected as the solution to the initial matrix. This solution signifies that if only one candidate is to be implemented, the X_k that provided the solution is the optimum candidate.

Table B-2 presents the solution to the initial matrix for this hypothetical case. In this example, X_2 is the variable selected and is considered to be implemented giving rise to the new reduced matrix illustrated in Table B-3. (Note, since X_2 is now considered selected, it no longer remains in the problem set.) During this process, improvement candidates may drop out of the reduced solutions because they are no longer economically viable due to non-linear effects. It is also possible that the variables previously screened out may enter into the reduced problem IP solution. This is illustrated in Table B-2 where X_3 entered the solution set even though it was not part of the original IP solution; X_1 and X_5 dropped out. This process continues until either the capital constraint is met or all of the viable alternatives have been accepted. Tables B-4 though B-7 illustrate these steps for the remainder of the hypothetical problem. In this example, the final solution is X_2, X_3, X_4, and X_6 with a capital cost of $976,900 and a total increase in revenue of $1,156,100. The non-linearities caused the total increase in revenue to be $26,900 less than the initial IP solution predicted. The algorithm was terminated by the cost constraint, i.e., all of the remaining options exceeded the remaining capital constraint, IC_3^*.

For those instances in which constraints are not exceeded or non-existent, the algorithm proceeds as described except the IP algorithm and constraints are not used. The only criterion for acceptance of the improvements is that each improvement remains beneficial after another component is improved.

V_k^*	$1,086,300	$1,141,100	$1,103,400	$1,137,500	$1,139,250
IC_k^*	$848,000	$638,000	$900,000	$770,000	$962,100
X_k	X_1	X_2	X_4	X_5	X_6
X_1	0	0	1	1	1
X_2	1	0	1	1	1
X_3	0	1	0	0	0
X_4	1	1	0	1	1
X_5	1	0	1	0	1
X_6	1	1	1	1	0
X_7	0	0	0	0	0

Solution is X_2

Table B-2. Initial DP Matrix Solution

V_k^*			
IC_k^*	$161,00	$538,00	$600,100
X_k	X_3	X_4	X_6
X_1			
X_3	0		
X_4		0	
X_5			
X_6			0
X_7			

Table B-3. First Reduced DP Matrix

V_k^*	$1,85,900	$1,259,600	$1,286,350
IC_k^*	$161,00	$538,00	$600,100
X_k	X_3	X_4	X_6
X_1	0	0	0
X_3	0	1	1
X_4	1	0	1
X_5	0	0	0
X_6	1	1	0
X_7	0	0	0

Solution is X_4

Table B-4. First Reduced DP Matrix Solution

V_k^*		
IC_k^*	$61,000	$500,100
X_k	X_3	X_6
X_1		
X_3	0	
X_5		
X_6		0
X_7		

Table B-5. Second Reduced DP Matrix

V_k^*	$1,115,050	$1,116,900
IC_k^*	$61,000	$500,100
X_k	X_3	X_6
X_1	0	0
X_3	0	1
X_5	0	0
X_6	1	0
X_7	0	0

Solution is X_6

Table B-6. Second Reduced DP Matrix

V_k^*	$432,000
IC_k^*	$23,100
X_k	X_3
X_1	0
X_3	0
X_5	0
X_7	0

Solution is X_3: No other candidates remain

Table B-7. Final Reduced DP Matrix