Scientific Research Publishing

American Journal of Operations Research, 2017, 7, 99-112
http://www.scirp.org/journal/ajor
ISSN Online: 2160-8849
ISSN Print: 2160-8830

Ant-Colony Optimization for the System Reliability Problem with Quantity Discounts

Patrick R. McMullen

School of Business, Wake Forest University, Winston-Salem, North Carolina, USA
Email: mcmullpr@wfu.edu

How to cite this paper: McMullen, P.R. (2017) Ant-Colony Optimization for the System Reliability Problem with Quantity Discounts. *American Journal of Operations Research*, **7**, 99-112.
https://doi.org/10.4236/ajor.2017.72007

Received: December 29, 2016
Accepted: March 12, 2017
Published: March 15, 2017

Abstract

This research presents an approach based upon ant-colony optimization to address the system reliability problem. For each component of a system, the number of units in parallel needs to be chosen to maximize the reliability for the entire system. As more parallel units are selected, costs increase in a proportional fashion. For this effort, quantity discounts for additional parallel units are considered, and the budget for purchase of parallel units is limited. Ant colony optimization methodology is employed to find an optimal system reliability that satisfies the budget constraint. The methodology is employed for several test problems, and near-optimal solutions are found.

Keywords

Heuristic, Ant-Colony Optimization, Search

1. Introduction

In the pursuit of system design and system engineering, it is important to construct a reliable system. One general way to do this is to include redundancies for each component. This is done to overcome failures of individual components. Each redundancy can be thought of as a backup unit. When one unit fails, a backup unit is activated, so that the component does not fail. The more redundancies, or backups a component has, the less likely the component is to fail. At the same time, however, the cost increases for each redundancy, or backup unit that is added to the component. As such, it is desired to construct a reliability system that maximizes system reliability while simultaneously adheres to a limited budget.

Unfortunately, problems of this nature are difficult to solve for a couple of reasons: they are combinatorial in nature, making it difficult to enumerate all possible solutions. Additionally, these problems are highly nonlinear, with a po-

lynomial order equal to the number of decision variables, thereby preventing the use of traditional mathematical programming approaches.

Because of the challenging mathematics associated with these types of problems, heuristic approaches can be employed to find reasonable, if not optimal solutions. Additionally, if the heuristic approach is well thought out, the approach can find near-optimal solutions with a reasonable computational effort. For this research effort, an ant-colony optimization approach is employed as a sort of traversal through each system component with the intent of finding a "traversal" that provides, for each component, a high overall-system reliability that meets a specified budget.

The subsequent sections of the paper present the details of the problem at hand, describe the heuristic methodology used, present a simple example problem, discuss applying the heuristic to some test problems, and compare these results to the optimal solution. General observations, and opportunities for subsequent research are then offered.

2. Background

The reliability of a system is the mathematical product of the effective reliability for each individual component. The effective reliability of an individual component is a function of the number of backup or parallel units. With more parallel units, the component's effective reliability increases. This principle is well-documented [1] [2] [3]. The effective cost also rises when more parallel units are added. Figure 1 details how the effective component reliability and effective cost works across three components—for each backup units/component combination, the first line shows reliability, while the second lines shows cost.

For this example, we assume the reliability for a single parallel unit for component j is R_j, where $0 < R_j < 1$. We also assume the cost for a single parallel unit for component j is C_j. The effective reliability for component j is $1-\left(1-R_j\right)^i$, where i is the number of units in parallel. The effective cost for component j is $C_j \sum_{h=0}^{i=1} D^h$, where D is the quantity discount factor ($0 < D \le 1$), with i units in parallel. The boxes in orange show the number of parallel units used for each component: three units for component (1), two units for component (2), and four units for component (3). The system reliability and system cost

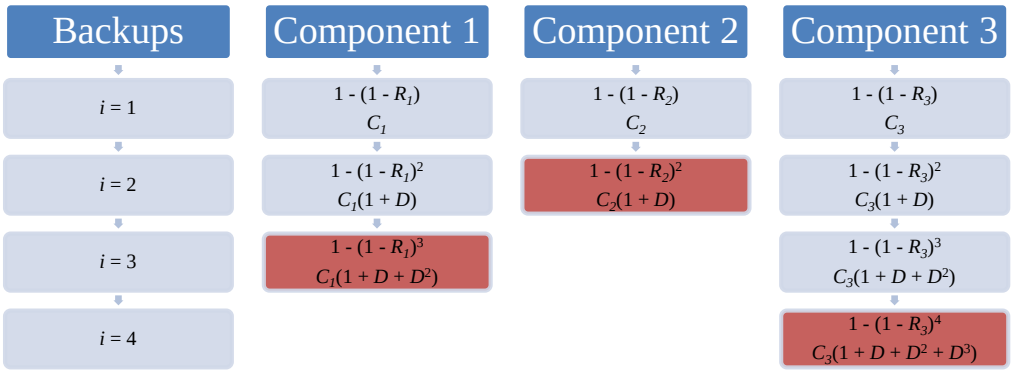

Figure 1. Components in series and parallel.

are shown as follows:

$$\text{System Reliability} = \left[1-\left(1-R_1\right)^3\right]\left[1-\left(1-R_2\right)^2\right]\left[1-\left(1-R_3\right)^4\right]$$

$$\text{System Cost} = \left[C_1\left(1+D+D^2\right)\right]+\left[C_2\left(1+D\right)\right]+\left[C_3\left(1+D+D^2+D^3\right)\right]$$

Selection of the number of units to employ for each component is the difficult part of this problem. Of course, we'd like to have as many parallel units as possible, as that increases the system reliability. Unfortunately, it also increases the system cost, and in reality, we are likely to have a limited budget. As such, we need to maximize reliability while staying within some specified budget.

Optimization via mainstream mathematical programming approaches is not possible here, due to the highly nonlinear property of the problem. Additionally, the problem has an enormous search space, rendering enumeration impractical. In order to address this problem, we need to select a search heuristic to find a desirable solution requiring a reasonable computational effort. While there are many heuristic search approaches possible, ant-colony optimization is the search approach for this particular research effort.

Ant-colony optimization is based on the concept of ants foraging for food. Ants essentially communicate with each other via secreting a substance called "pheromone." Pheromone attracts other ants. A trail rife with pheromone will likely induce other ants to follow its path, while at the same time, there is a relatively small probability for subsequent ants to deviate from this established path. This concept is illustrated by Figure 2. Initially, ants are confined to their nest ($t = 0$). At $t = 1$, the ants pursue the food in what is seen as random fashion. At $t = 2$, there emerges a pattern where the ants start to enforce a direct route to the food. At steady state ($t = \infty$), the ants have seemingly found the shortest (best) route to the food, but the possibility of finding alternate routes still exists [4] [5].

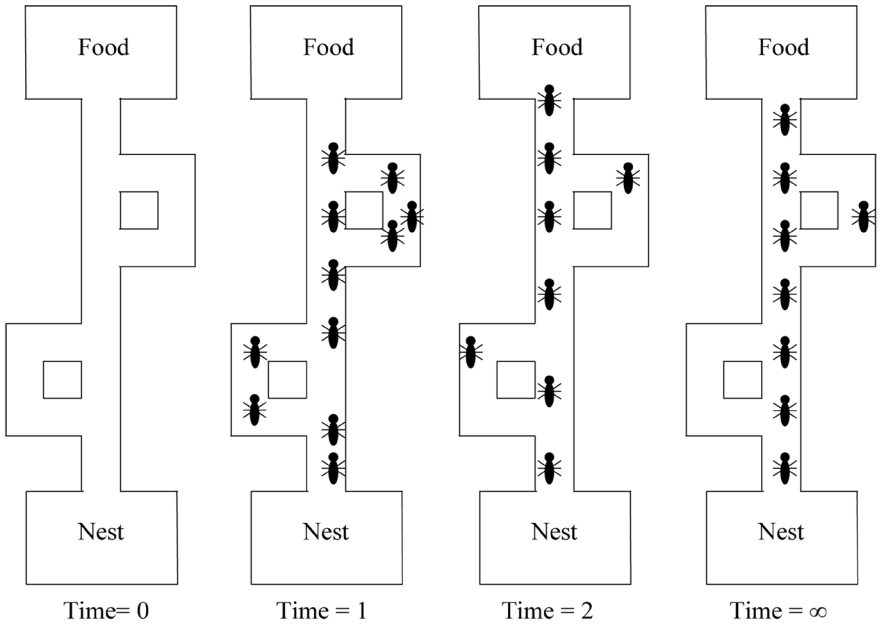

Figure 2. Traversal of ants [6].

For this research effort, we will simulate the behavior of the ants so that they traverse the individual components of a system with the intent of maximizing system reliability within budget-that is, having ants finding the "best" path during their traversals. Dealing with a limited budget and quantity discounts is a unique feature of this effort [7]. This is analogous to ants finding the most efficient route to a food source. The next section formalizes the search process.

3. Methodology

Here, we first formulate the general problem, detail the search heuristic, and offer a simple example to illustrate the methodology.

3.1. Formulation

Before presenting the formulation, the following terms in Table 1 are introduced.

For this problem, our objective is to maximize the system reliability-the product of all component reliabilities. Our individual component reliabilities are dependent on the number of parallel units used for each component, which is a decision variable (x_j). The number of parallel units used for each component determines the cost of each individual component. More units used in parallel for each component increases the cost for the component, subsequently increasing the total cost. The total cost is limited by the budget, B. From a mathematical standpoint, we can represent our model in mathematical programming form via the following objective function and constraint, where decision variables are specified as integer, and in the range [1, n]:

$$\text{Max}: R_s = \prod_{j=1}^{m}\left(1-\left(1-R_j\right)\right)^{x_j} \tag{1}$$

$$\text{Subject to}: \sum_{j=1}^{m}\left(C_j \sum_{h=1}^{x_j} D^{(h-1)}\right) \le B \tag{2}$$

$$x_j \text{ integer}, \forall j \tag{3}$$

$$x_j \ge 1, \forall j \tag{4}$$

$$x_j \le n, \forall j \tag{5}$$

Table 1. Data for example problem.

Term	Definition
x_j	number of backup units for component j (decision variable)
n	number of possible backup units for any component
m	number of components in system
R_j	base reliability of component j
C_j	base cost of component j
R_s	system reliability
D	discount factor
B	budget

From inspection of the above model it is clear that our objective function is highly nonlinear, with a polynomial order of m. Additionally, our budget constraint is not continuous. Because of this, traditional mathematical programming approaches do not guarantee optimality. As such, we need to pursue a heuristic to provide a reasonable optimal solution via a reasonable computational effort.

Several heuristic approaches are available to address this type of problem. A genetic algorithm can be used to address this problem. Genetic Algorithms have proven effective in solving difficult optimization problems. The down side of Genetic Algorithms is computational inefficiency. A Genetic Algorithm generates binary solutions, performs analyses, manipulates the binary characteristics of the problem, converts binary solutions back into decimal solutions, and repeats this process until some pre-specified stopping criterion is met. This puts a lot of strain on the processor, resulting in a protracted search.

For this research effort, an artificial agent approach is used-something similar to ant colony optimization. An "ant" will traverse each component, starting with component 1 and concluding with component m. The choice of how many parallel units to select for each component during the traversal is probabilistic in nature, and proportional to the relative desirability of component i for each component j. For each unique traversal, the system reliability is computed, and if feasible, the system reliability is compared to the best system reliability found thus far. If the new solution is found to be the best so far, it becomes the new best solution, and the attractiveness of the solution components are enhanced accordingly. If the new solution is not the best found thus far, the attractiveness of the solution is diluted accordingly.

3.2. Ant-Colony Heuristic

Prior to detailing the optimization algorithm, the following terms in Table 2

Table 2. Data for example problem.

Term	Definition
x_j	number of backup units for component j (decision variable)
n	number of possible backup units for any component
m	number of components in system
i	backup unit index $(i=1,\ 2,\cdots,n)$
j	component index $(j=1,2,\cdots,m)$
r_{ij}	reliability of i parallel units for component j
c_{ij}	cost of i parallel units for component j
$pher_{ij}$	pheromone associated with i parallel units for component j
imp_{ij}	number of times i parallel units for component j has been in the best solution
$prob_{ij}$	probability of selecting i parallel units for component j
r_s	system reliability
c_s	system cost
$maxR$	highest obtained value of system reliability
A	pheromone amplifier
D	discount factor
B	budget

are defined:

3.2.1. Step 1: Initialization

First of all, the reliability and cost matrices are determined. The reliability of associated with employing i parallel units for component j is determined as follows:

$$r_{ij} = \left(1-\left(1-R_i\right)\right)^i, \forall j \tag{6}$$

The cost associated with employing i parallel units for component j is determined as follows:

$$c_{ij} = C_j \sum_{h=1}^{i} D^{(h-1)}, \forall i, j \tag{7}$$

Because we desire high values of reliability and low values of cost when choosing the number of parallel units for each component, pheromone is a function of the ratio of reliability to cost for each parallel unit/component combination. As such, pheromone is as follows:

$$\text{pher}_{ij} = r_{ij} / c_{ij}, \forall i, j \tag{8}$$

$$\text{imp}_{ij} = 1, \forall i, j \tag{9}$$

Our level of pheromone, along with the initialized improvement matrix, is then used to determine the probability of selecting i parallel units for component j. This is determined as follows:

$$\text{prob}_{ij} = \frac{\left(\text{pher}_{ij}^{\alpha}\right)\left(\text{imp}_{ij}^{\beta}\right)}{\sum_{i=1}^{n}\left(\left(\text{pher}_{ij}^{\alpha}\right)\left(\text{imp}_{ij}^{\beta}\right)\right)}, \forall j \tag{10}$$

The probability calculation above combines both pheromone and the number of overall best-solution improvements associated with selecting i parallel units for component j. The values of α and β are amplifiers, or weights for pheromone and improvements, respectively. Max$R$ is initialized to zero. Thus concludes the initialization process. The traversal process is now described.

3.2.2. Step 2: Traversal

An artificial agent, or "ant" traverses all m components, and at each component, a number of backup units are selected. We use the decision variable x_j to represent the number of backup units selected for component j. This selection is done via Monte-Carlo simulation using the probabilities shown in Equation (10). A uniformly-distributed random number on the [0, 1] interval is generated and the appropriate value of x_j is determined by where this random number lies in accordance with Equation (10). At the completion of the traversal, we have a solution.

3.2.3. Step 3: Assessment

Our traversal of the m components provides us with a solution, where x_j represents the number of parallel units employed for component j. We need to determine the system reliability and system cost of our recently-obtained solu-

tion. System reliability is obtained as follows:

$$r_s = \prod_{j=1}^{m} r_{x_j,j}, \forall j \tag{11}$$

System cost is defined as follows:

$$c_s = \sum_{j=1}^{m} c_{x_j,j}, \forall j \tag{12}$$

If the system reliability obtained via Equation (11) exceeds Max$R$, and the system cost obtained via Equation (12) is less than the budgeted amount (B), the solution is the best one found thus far, and this solution becomes the best solution thus far:

$$\text{Max}R = r_s \tag{13}$$

3.2.4. Step 4: Pheromone Update

If the recently found solution is the best thus far as defined above, the pheromone is updated accordingly:

$$\text{pher}_{x_j j} = \text{pher}_{x_j j} + A, \forall j \tag{14}$$

Additionally, the number of improvements associated with this new best solution is incremented as follows:

$$\text{imp}_{x_j j} = \text{imp}_{x_j j} + 1, \forall j \tag{15}$$

Otherwise-that is, if the new solution is not an improvement over the best-found solution thus far, the pheromone is updated accordingly.

$$\text{pher}_{x_j j} = \text{pher}_{x_j j} - A, \forall j \tag{16}$$

Equations (14) and (15) *enhance* a solution that was found to be desirable-the characteristics of this solution are made more likely to occur in future traversals. Equation (16) *dilutes* a solution that was not found to be desirable—the characteristics of this solution are made less likely to occur in future traversals. Regardless of the type of pheromone adjustment made above, probabilities must be re-adjusted for the next traversal. This is done via Equation (10).

3.2.5. Step 5: Repeat

Steps 2, 3 and 4 are repeated until some stopping condition is reached, which is typically a user-defined condition. The last reported "best solution" reported is considered the best solution.

3.3. Example

To illustrate the presented methodology, we present a small example. Consider a problem with eight components (m = 8) and up to six units in parallel (n = 6) are permitted for each component. Table 3 below shows the given R_j and C_j values.

Table 4 shows the reliability matrix (r_{ij} values) for up to six units in parallel for each of the eight components. Equation (6) was used to determine these values. Please note that for each individual component, the reliability increases with

Table 3. Data for example problem.

	1	2	3	4	5	6	7	8
R_j	0.885	0.90	0.92	0.89	0.93	0.925	0.92	0.97
C_j	$ 7.50	$3.50	$6.00	$4.00	$9.00	$6.00	$5.50	$8.00

Table 4. Reliability matrix.

	1	2	3	4	5	6	7	8
1	0.885000	0.900000	0.920000	0.890000	0.930000	0.925000	0.920000	0.970000
2	0.986775	0.990000	0.993600	0.987900	0.995100	0.994375	0.993600	0.999100
3	0.998479	0.999000	0.999488	0.998669	0.999657	0.999578	0.999488	0.999973
4	0.999825	0.999900	0.999959	0.999854	0.999976	0.999968	0.999959	0.999999
5	0.999980	0.999990	0.999997	0.999984	0.999999	0.999999	0.999997	1.000000
6	0.999998	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000	1.000000

each additional unit in parallel. That is, the more units used in parallel, the higher the component's reliability. As an example, if three parallel units are used for component six, the component reliability is 0.999578 (r_{36} = 0.999578).

Table 5 shows the cost matrix for i units in parallel for each component (c_{ij} values). These values are determined by Equation (7), along with a discount factor (D) of 0.97. Note that for each component, using more units in parallel increases the component cost by a factor of D. As such, there is a tradeoff with using more units in parallel—the more units used in parallel, the higher the cost. Because of this, one must carefully choose the number of parallel units to optimize the system reliability with a limited budget, B. As an example, if we use three parallel units for component two, our cost is $10.19 ($r_{32}$ = 10.19).

As an agent or "ant" traverses the m components of the system, they must make a decision on how many parallel components to employ for the next component. This is a stochastic process, and the level of pheromone is influential in the selection process. For this research effort, the level of pheromone is determined via Equation (8), which is the ratio of r_{ij} to c_{ij} for all i,j combinations. It should be noted that since the imp_{ij} values are initialized to one, they do not affect the probabilities at this point in the search.

For this example, for three parallel units being used for component one is $0.0457\left(\left[\text{pher}_{ij}\right]\left[\text{imp}_{ij}\right]=0.0457\right)$. Additionally, the values of and α and β and both assumed to equal one. These values are shown in Table 6. For each component, the total amount of $(\text{pher}_{ij})(\text{imp}_{ij})$ is summed on the bottom line.

These values are converted to probabilities via Equation (10), and are shown in Table 7. For each individual component, one will note that the sum of probabilities is unity. Monte Carlo simulation is used to select the number of parallel units employed for each component. For example, there is a 0.1057 probability that there will be four parallel units employed for component eight (prob_{48} = 0.1057).

For the purpose of our example, let us assume that we have a discount factor

Table 5. Cost matrix.

	1	2	3	4	5	6	7	8
1	$7.50	$3.50	$6.00	$4.00	$9.00	$6.00	$5.50	$8.00
2	$14.78	$6.90	$11.82	$7.88	$17.73	$11.82	$10.84	$15.76
3	$21.83	$10.19	$17.47	$11.64	$26.20	$17.47	$16.01	$23.29
4	$28.68	$13.38	$22.94	$15.29	$34.41	$22.94	$21.03	$30.59
5	$35.32	$16.48	$28.25	$18.84	$42.38	$28.25	$25.90	$37.67
6	$41.76	$19.49	$33.41	$22.27	$50.11	$33.41	$30.62	$44.54

Table 6. $(pher_{ij})(imp_{ij})$ matrix.

	1	2	3	4	5	6	7	8
1	0.1180	0.2571	0.1533	0.2225	0.1033	0.1542	0.1673	0.1213
2	0.0668	0.1436	0.0841	0.1254	0.0561	0.0841	0.0917	0.0634
3	0.0457	0.0981	0.0572	0.0858	0.0382	0.0572	0.0624	0.0429
4	0.0349	0.0747	0.0436	0.0654	0.0291	0.0436	0.0475	0.0327
5	0.0283	0.0607	0.0354	0.0531	0.0236	0.0354	0.0386	0.0265
6	0.0239	0.0513	0.0299	0.0449	0.0200	0.0299	0.0327	0.0225
Sum	0.3177	0.6855	0.4035	0.5970	0.2702	0.4044	0.4402	0.3093

Table 7. Probability matrix.

	1	2	3	4	5	6	7	8
1	0.3715	0.3751	0.3800	0.3727	0.3824	0.3812	0.3800	0.3920
2	0.2103	0.2095	0.2083	0.2100	0.2077	0.2080	0.2083	0.2050
3	0.1440	0.1430	0.1418	0.1437	0.1412	0.1415	0.1418	0.1388
4	0.1098	0.1090	0.1080	0.1095	0.1075	0.1078	0.1080	0.1057
5	0.0891	0.0885	0.0877	0.0889	0.0873	0.0875	0.0877	0.0858
6	0.0754	0.0749	0.0742	0.0752	0.0739	0.0740	0.0742	0.0726
Sum	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

of $D = 0.97$, with a budget of $B = \$200$. We also assume Monte Carlo simulation resulted in a solution of $x = [3, 4, 3, 3, 2, 3, 2, 2]$. That is three parallel units for component one, four parallel units for component two, etc. Tables 4-6 are enhanced to show the associated component reliabilities and costs associated with this solution-these values are shaded. With this solution, we compute the system reliability as follows:

$$r_s = (0.998479)(0.999900)(0.999488)(0.998669)$$
$$(0.995100)(0.999578)(0.993600)(0.999100) = 0.984008.$$

The system cost is determined as follows:

$$c_s = \$21.83 + \$13.38 + \$17.47 + \$11.64 + \$17.73 + \$17.47 + \$10.84 + \$15.76 = \$126.11$$

Given that this is our first solution, it provides us with the maximum system reliability ($\max R$) thus far. Our system cost is also within our budget of \$200 ($c_s <$

B). As such, this is our best solution thus far. The value of Max$R$ is replaced with r_s. Because this solution is an improvement over the previous best solution, the pheromone must be updated accordingly. Table 8 and Table 9 show how the pheromone matrix is updated by the pre-selected amplifier value equal to 0.01 (A = 0.01) for the appropriate solution values-affected values are shaded. Non-affected values are the same as before.

Because this solution is the best found thus far, our improvement matrix is also updated in accordance with Equation (15).

Because the pheromone and improvement matrices have been adjusted, the corresponding probability matrix must also be adjusted. All probability values will change. Probabilities associated with the recently-obtained best solution will increase, while the other probabilities will decrease—this is the result of enhancing the pheromone and improvement values for the recently found best solution. Table 10 shows the updated probability matrix.

Table 8. Updated pheromone matrix.

	1	2	3	4	5	6	7	8
1	0.1180	0.2571	0.1533	0.2225	0.1033	0.1542	0.1673	0.1213
2	0.0668	0.1436	0.0841	0.1254	0.0661	0.0841	0.1017	0.0734
3	0.0557	0.0981	0.0672	0.0958	0.0382	0.0672	0.0624	0.0429
4	0.0349	0.0847	0.0436	0.0654	0.0291	0.0436	0.0475	0.0327
5	0.0283	0.0607	0.0354	0.0531	0.0236	0.0354	0.0386	0.0265
6	0.0239	0.0513	0.0299	0.0449	0.0200	0.0299	0.0327	0.0225

Table 9. Updated improvement matrix.

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	1	2	2
3	2	1	2	2	1	2	1	1
4	1	2	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1

Table 10. Updated probability matrix.

	1	2	3	4	5	6	7	8
1	0.3079	0.3295	0.3189	0.3165	0.2982	0.3202	0.3031	0.3089
2	0.1743	0.1841	0.1750	0.1784	0.3816	0.1746	0.3685	0.3738
3	0.2906	0.1257	0.2796	0.2726	0.1103	0.2791	0.1131	0.1092
4	0.0911	0.2171	0.0907	0.0930	0.0840	0.0905	0.0861	0.0833
5	0.0738	0.0778	0.0736	0.0755	0.0681	0.0735	0.0699	0.0675
6	0.0624	0.0658	0.0622	0.0639	0.0577	0.0621	0.0592	0.0573
Sum	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

The iterative process: traversal, assessment, and updating of pheromone, improvements and probabilities continues until some stopping criterion is met. For this simple example, 1000 iterations are used, resulting in the solution $x = [5, 5, 4, 6, 4, 4, 4, 3]$, with a system reliability of $r_s = 0.999804$, and a system cost of $c_s =$ \$198.68. This solution was also determined to be optimal via enumeration of all solutions.

4. Experimentation

The presented methodology is used on four test problems. Details of these test problems are shown in Table 11. The number of possible solutions is equivalent to the following:

$$\text{Solutions} = m^n = \text{Backups}^{\text{Components}}$$

It should be noted that using more than eight parallel units (backups) for each component results in diminishing utility-system reliability gains become negligible, and only the cost increases, leading to suboptimal solutions.

For each of the test problems, the R_j and C_j values are provided in Tables 12-15. This information enables us to pursue the presented heuristic, in our effort to find good solutions for these test problems.

Table 11. Test problems.

Problem	Backups	Components	Budget (B)	Discount (D)	Solutions
ACO-1	8	10	275	0.95	1,073,741,824
ACO-2	8	12	450	0.99	68,719,476,736
ACO-3	8	13	400	0.97	549,755,813,888
ACO-4	8	14	650	0.95	4,398,046,511,104

Table 12. Test problem ACO-1.

	1	2	3	4	5	6	7	8	9	10
R_j	0.7516	0.7195	0.8059	0.6875	0.7751	0.8373	0.9081	0.7877	0.7275	0.8236
C_j	6.34	6.00	7.10	5.54	6.81	4.91	5.38	6.75	5.73	6.44

Table 13. Test problem ACO-2.

	1	2	3	4	5	6	7	8	9	10	11	12
R_j	0.681	0.7896	0.8488	0.8860	0.7652	0.6293	0.7834	0.7526	0.8239	0.9543	0.7427	0.8465
C_j	8.30	6.30	9.70	7.55	9.74	4.85	4.50	6.80	9.74	4.00	7.13	7.80

Table 14. Test problem ACO-3.

	1	2	3	4	5	6	7	8	9	10	11	12	13
R_j	0.885	0.900	0.920	0.890	0.930	0.925	0.920	0.970	0.925	0.900	0.750	0.780	0.912
C_j	7.50	3.50	6.00	4.00	9.00	6.00	5.50	8.00	4.50	7.50	6.50	7.90	8.30

Table 15. (a) Test Problem ACO-4. (b) Test Problem ACO-4.

(a)

	1	2	3	4	5	6	7
R_j	0.827	0.842	0.865	0.861	0.885	0.898	0.829
C_j	4.28	5.25	11.03	13.34	9.49	9.68	16.31

(b)

	8	9	10	11	12	13	14
R_j	0.839	0.853	0.927	0.828	0.838	0.823	0.871
C_j	8.36	9.37	4.93	11.64	11.16	11.93	13.28

5. Experimental Results

Table 16 shows the results of the heuristic approach as compared to the optimal solution as obtained via complete enumeration. For all problems, the search parameters for α and β were set to 1 and 1.5 respectively. For each problem, the ant-colony optimization heuristic was performed using the number of iterations specified in Table 16. For each problem, the heuristic was repeated ten times, and the mean and standard deviation of the objective function value were calculated. From inspection of Table 16, it should be noted that the heuristic performed, on average, no worse than 0.03% inferior to optimal.

Figure 3 shows a "snapshot" of the improvement matrix for ACO-4. Each column represents one of the (14) components, while each row represents the number of units used in parallel for the component of interest. The intensity of each circle represents the frequency that i parallel units were used for component j when a new "best solution" was found. The darker the circle, the more often that particular number of parallel units was found to be beneficial for the best solution. For example, (7) parallel units is a desirable number value for component (1), while (1) parallel unit was not desirable for component (1). This improvement matrix is used in the probability calculations to "reinforce" desirable solutions for subsequent ant traversals.

6. Concluding Comments

We have presented an ant-colony optimization approach to address a system reliability problem with quantity discounts and a constrained budget. Because conventional optimization approaches are not feasible for such large, nonlinear problems, we must pursue search heuristics. Of course, ant-colony optimization is one of many options. Tabu Search, Simulated Annealing and Genetic Algorithms are possible search candidates. For this type of problem, however, ant-colony optimization serves us well due to the fact that we are able to model our desire for a solution by simulating an ant traversing the search space in a logical fashion. This would be more difficult with the aforementioned approaches. Furthermore, these other search approaches can be computationally expensive due to the management details associated with these approaches.

Of course, there are always opportunities for additional research, and this effort

Table 16. Heuristic solutions compared to optimal.

	Optimal Solution		Heuristic Solution		
Problem	Solution (x_j)	Reliability	Iterations	Reliability ($\bar{x}$)	Reliability (s)
ACO-1	[5, 6, 4, 7, 5, 4, 3, 5, 6, 5]	99.380807%	25,000	99.351834%	0.166855%
ACO-2	[7, 5, 4, 4, 6, 8, 6, 6, 5, 3, 6, 4]	99.659840%	50,000	99.638339%	0.046367%
ACO-3	[5, 5, 5, 5, 4, 4, 5, 3, 5, 5, 8, 7, 5]	99.980674%	100,000	99.974600%	0.006069%
ACO-4	[6, 5, 5, 5, 5, 4, 5, 6, 5, 4, 6, 5, 6, 5]	99.918462%	250,000	99.906818%	0.007980%

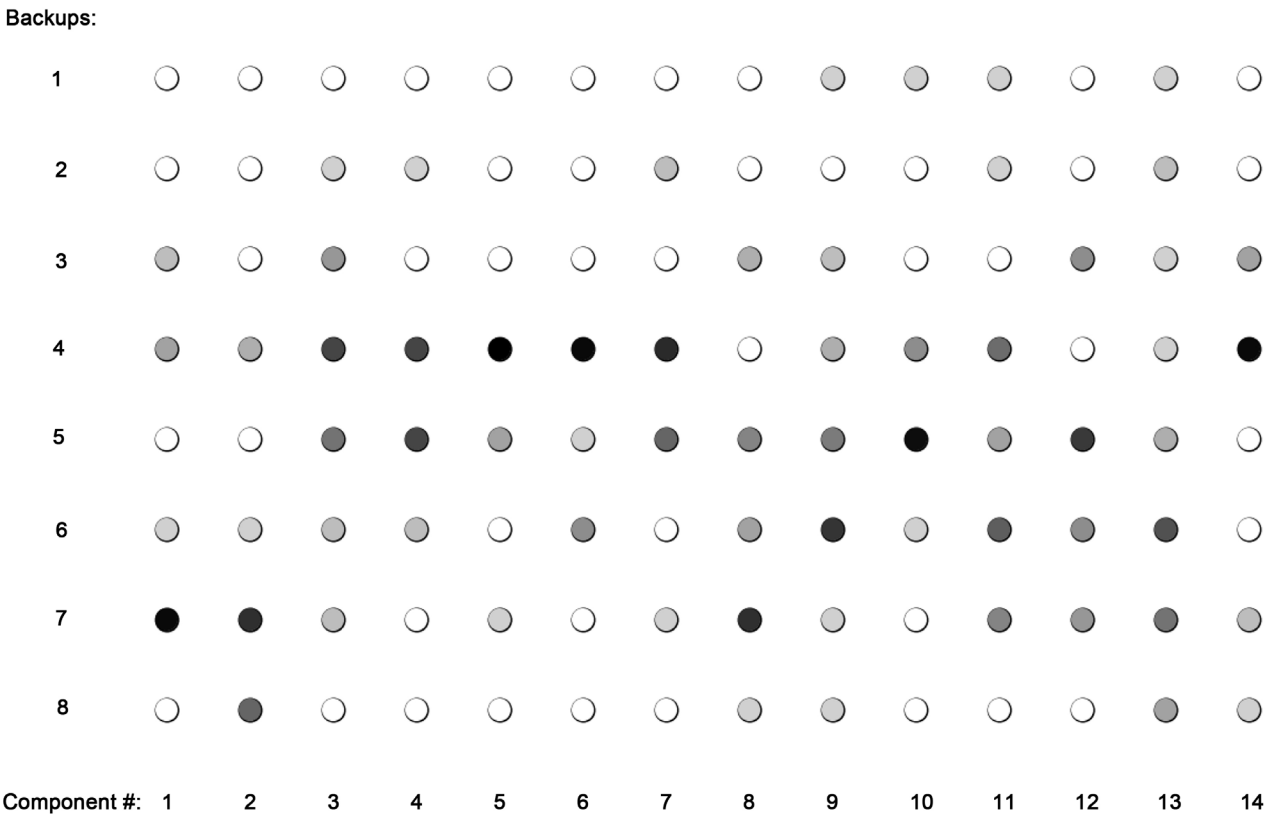

Figure 3. Snapshot of improvement matrix for ACO-4.

is no exception. The largest problem addressed has 4.4 trillion possible solutions. Comparing our best solution found via the heuristic to the optimal solution with 4.4 trillion possibilities required some work, but with today's computational resources, it was surprisingly possible. In fact, the size of the experimental problems is considered one of the unique features associated with this effort—other efforts work with are unable to provide solutions to problems of this practical size [8] [9]. To this end, larger problems can be pursued in the future. Additionally, other heuristic approaches can be tried, such as those aforementioned.

References

[1] Neubeck, K. (2004) Practical Reliability Analysis. Prentice Hall, New Jersey

[2] O'Connor, P.D.T. (2002) Practical Reliability Engineering. Fourth Edition, John Wiley & Sons, New York.

[3] Todinov, M. (2016) Reliability and Risk Models: Setting Reliability Requirements. Wiley, New Jersey.

[4] Dorigo, M., Di Caro, G. and Gambardella, L. M. (1999) Ant Algorithms for Discrete Optimization. *Artificial Life*, **5**, 137-172. https://doi.org/10.1162/106454699568728

[5] Dorigo, M., Maniezzo, V. and Colorni, A. (1996) The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, **26**, 29-41. https://doi.org/10.1109/3477.484436

[6] McMullen, P.R. and Tarasewich, P (2003) Using Ant Techniques to Solve the Assembly Line Balancing Problem. *IIE Transactions*, **35**, 605-617. https://doi.org/10.1080/07408170304354

[7] Liang, Y. and Smith, A. (2007) The Ant Colony Paradigm for Reliable Systems Design. *Computational Intelligence in Reliability Engineering*, **40**, 1-20. https://doi.org/10.1007/978-3-540-37372-8_1

[8] Thanitakul, P., Sa-ngiamvibool, W., Aurasopon, A. and Pothiya, S. (2013) Improved Ant Colony Optimization for Solving Reliability Redundancy Allocation Problems. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, **7**, 314-319.

[9] Luo, S., Cheng, L., Ren, B. and Zhu, Q. (2014) An Improved Intelligent Ant Colony Algorithm for the Reliability Optimization Problem in Cyber-Physical Systems. *Journal of Software*, **9**, 20-25.