Development of a Generalized Framework for Aggregate Blending in Pavement Construction

Abstract

Blending of aggregates can be defined as the process of combining aggregates from different stockpiles to meet a specific range of sieve size distribution. Different techniques have evolved over time for solving such problems which ranges from simple graphical techniques to non-linear programming methods.

This study is aimed at proposing a generalized framework to find all possible solutions for any given number of stockpiles, aggregate sizes, and specification limits. Due to the constraint of large computation memory, the method is best suited for combining up to six different stockpiles. The solution also incorporates identification of the proportions closest to the mid-point of the gradation limits. In addition, a simple formulation for identifying proportions corresponding to minimum cost function is also presented. The applicability of the method has been demonstrated using a suitable numerical example involving four stockpiles. The final output of the study provides the user with three benefits during aggregate blending: a) identification of all possible solutions for the given number of stockpiles, b) identification of proportions closest to the mid-point gradation, and c) identification of the proportions corresponding to minimum cost function.

Keywords: *Aggregate blending, Mix-proportioning, Cost optimization, Pavement Construction*

Introduction

Aggregate blending is one of the basic steps in any mix design process related to pavements. Mineral aggregates in quarries are separated in varying sized fraction and are stockpiled. Various transportation agencies have their own requirement of the sieve size distribution of aggregates depending on its intended purpose. Aggregates used in the base and sub-base layers of asphalt pavements are coarser in comparison to the gradation used in the surface layer. The nominal maximum aggregate size also depends on the type of layer and its intended purpose. The process of aggregate proportioning has evolved from the basic understanding of Fuller's maximum density line of combining aggregates of different sizes to achieve the requirement of design voids in the aggregate structure (Roberts et al. 1996). Broadly, such gradations can be quantified as well graded, gap-graded, and uniformly graded. Aggregate gradation is graphically defined using the sieve size distribution plot using a semi-log scale with logarithmic of sieve sizes as the abscissa and percent passing as the ordinate (Murthy 2003).

Aggregate blending can be briefly defined as the process of combining various stockpiles to meet the desired sieve size distribution of the final blend (Easa and Can 1985b). The aggregate proportioning starts with three basic inputs: a) the number of stockpiles which has to be combined, b) the number of sieve sizes involved for the proportioning, and c) the target upper and lower limits of percent passing the various sieve sizes. Since the target is a range which needs to be satisfied, there can be various such combinations of the given stockpiles that can satisfy the required gradation limits (Kikuchi et al. 2012). Hence multiple solutions can be arrived at while blending of aggregates from different stockpiles.

A review of previous literature indicates that the process of aggregate blending can be broadly classified into three categories (Easa and Can 1985a; b; Kikuchi et al. 2012; Lee and Olson 1983; Neumann 1964; Ritter and Shaffer 1962; Toklu 2005): Graphical Methods; Trial and Error process, and Optimization Techniques. Triangular Chart Method, Asphalt Institute Method, and Routhfutch Method are among the popular graphical methods. The graphical methods offers simplicity to the process of blending and are still utilized by Engineers for blending two to three stockpiles. However, as the number of aggregate sizes and stockpile increases, the use of graphical methods becomes complicated. For example, Asphalt Institute graphical method and the triangular chart method cannot accommodate more than two or three stockpiles of aggregates respectively. The solutions obtained from the graphical methods can be further refined to meet the desired gradation range by use of trial and error process. Both the methods, i.e. graphical and trial and error method offers large deviations and are approximate solutions. Moreover, these methods cannot be used for cost optimization. Such complex aggregate blending problems involving more number of stockpiles and cost constraints can be solved using mathematical approaches such as linear and non-linear programming techniques (Easa and Can 1985a; Neumann 1964). Various optimization techniques, proposed by different researchers, ranges from simple least-square (LS) method to more robust models including quadratic programming (Easa and Can 1985a), genetic algorithms (Toklu 2005), and fuzzy optimization (Kikuchi et al. 2012). The methods have been applied to satisfy the conditions imposed on the objective function. Such objective function are usually defined to achieve the mid-point of the specification limits, reduction of cost and satisfaction of various physical parameters such as plasticity index and fineness modulus (Easa and Can 1985b; Neumann 1964; Ritter and Shaffer 1962). Most of the previous studies aim at identifying the 'best' combination of stockpiles to reach the mid-point of the desired aggregate range. This is done by minimizing the deviation in terms of least square error. However, the mid-point range is not necessarily the 'best' result for a given aggregate gradation. For example, a designer, depending on the project requirement and his expert judgment may consider getting a stockpile combination giving the highest amount of fines within the given range. Hence it is more logical and appropriate to have all possible combinations of stockpiles satisfying the aggregate gradation criteria and then decide which of the combination to choose.

In this study, an attempt has been made to develop an algorithm to find out all possible combination of stockpiles that can satisfy the desired aggregate range criteria corresponding to various sieve sizes. In addition, the program also identifies the blending proportion based on the two parameters: closest to the mid-point of the desired specification limits and minimization of cost function. The method is applicable to any form of aggregate gradation, irrespective of the use of these aggregates in any specific layer of pavement. The algorithm has been converted to an easy to use program which can handle any number of aggregate sieve sizes, up to six combinations of stockpiles. It is usually seen that not more than six stockpiles are used for a project involving the construction of pavements. However, the algorithm proposed can be used for any number of stockpiles.

Model Development

In this section, the algorithm for the development of aggregate blending model is described. The algorithm commences with prompting the user to input the preliminary information needed to construct the basic matrices required. The procedure starts with asking the user to enter the number of sieves, M, as used in the sieve analysis process. This is followed by entering the number of available stock piles, N, required for mix-proportioning to satisfy the target gradation.

The sieves are chosen based on the standard required gradation as outlined by the respective transportation agency. The pre-requirement to make the calculation involves sieve size analysis of each stockpile separately and noting the weight retained (WR) in various sieves involved. This information is used to calculate the percentage passing matrix $[\alpha]$ that corresponds to the percent passing various sieves for each of the stockpiles. The percent passing is calculated as follows:

$$\alpha_{kj} = \frac{\sum WR_{(M-k)j}}{\sum Total\ Weight_j} \times 100 \quad (1)$$

In equation (1), M is the number of sieves involved. $\sum WR_{(M-k)j}$ indicates the summation of weight retained in all the other sieves smaller than sieve size k for the j^{th} stockpile. For a given number of stockpiles, N, the order of the matrix will be $M \times N$ and is denoted as $[\alpha_{M\times N}]$. The aggregate gradation matrix $[\beta]$ is constructed such that it corresponds to the lower and upper bounds of the target gradation as entered by the user. The matrix is of the order of $M \times 2$ and is represented as $[\beta_{M\times 2}]$. The integer solution matrix $[\gamma]$, which stores all the integer solutions to the summation equation corresponding to the number of stockpiles, N, is pre-calculated and stored in a database. Hence, as N is entered by the user, the required matrix is loaded from the stored workspace. This matrix is a two dimensional matrix of order $T \times N$, where T is the total number of integer solutions to the summation equation and is represented as $[\gamma_{ij}]$, where i is the i^{th} solution to the summation equation (2) and j is the proportion of the j^{th} stockpile corresponding to that particular solution. All these are the integer solutions of the summation equation which is shown as:

$$\sum_{i=1}^{i=N} x_i = 100 \qquad\qquad 100 \geq x_i \geq 0 \tag{2}$$

This model is designed for N ranging from 2 to 6. Thus, there are five matrices that are pre-calculated and stored separately in a database. The total number of integer solutions, T, for the summation equation (2) above can be calculated by the following formula:

$$T = \binom{100+N-1}{N-1} \tag{3}$$

where,

$$\binom{n}{r} = \frac{n(n-1)\ldots(n-r+1)}{r(r-1)\ldots 1} \tag{4}$$

Table 1 shows the calculated value of T as a function of N.

Table 1. Total number of integer solutions to the summation equation

Value of N	Calculated T for given N
2	101
3	5151
4	176,851
5	4,598,126
6	96,560,646

For instance, as per equation (3) and Table 1, if the number of stockpiles is 3, the matrix called for calculation in the program will be of the order of 5151×3 as shown below:

$$[\gamma_{ij}] = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \gamma_{1,3} \\ \vdots & \vdots & \vdots \\ \gamma_{5151,1} & \gamma_{5151,2} & \gamma_{5151,3} \end{bmatrix} \tag{5}$$

In the above matrix, $\gamma_{i,1} + \gamma_{i,2} + \gamma_{i,3} = 100 \; \forall \; i = 1 \; to \; 5151$

Calculations

The first calculation which is performed, post saving the initial inputs is the percent passing of the combined proportion of i, for each sieve size k. This calculation is done for all the possible combinations, i.e., for each row of $[\gamma_{T\times N}]$. The operation done is as follows:

$$[\delta_{T\times M}] = ([\gamma_{T\times N}] \times [\alpha_{M\times N}]^{\mathrm{T}}) \div 100 \tag{6}$$

The $[\delta]$ matrix stores the combined contribution of all the stockpiles corresponding to the combined percent passing for each of the sieves entered for all the possible solutions T.

The next step is to start a search algorithm that checks for the satisfaction of the bound constraints entered by the user earlier as per the requirement. Thus, every row of the $[\delta]$ matrix is checked for the bound constraint and stored in the $[\beta_{N\times 2}]$ matrix. All the indices of the solutions satisfying the bound constraints are stored separately in a $\{\theta_S\}$ vector, where S is the number of integer solutions satisfying the aggregate gradation bound constraints.

The search algorithm is done in a single line of code, which can be explained in a number of steps as shown below:

- The algorithm creates two binary matrices of the order $T \times M$. These matrices are constructed by comparing the values of each of the integer solutions to the bound constraints for every sieve. Let these matrices be called $[\tau^1]$ and $[\tau^2]$. A row in $[\tau^1]$ will be all 1s if gradation calculated for all the sieves is greater than or equal to the respective lower bounds. Similarly, a row in $[\tau^2]$ will be all 1s if gradation calculated for all the sieves is lesser than or equal to the respective upper bounds.
- The binary assignment (0 or 1) is done as shown in equations (7) and (8):

$$\tau_{ij}^{1} = \begin{cases} 0 & \delta_{ij} < \beta_{1j} \\ 1 & \delta_{ij} \geq \beta_{1j} \end{cases} \tag{7}$$

$$\tau_{ij}^{2} = \begin{cases} 0 & \delta_{ij} > \beta_{2j} \\ 1 & \delta_{ij} \leq \beta_{2j} \end{cases} \tag{8}$$

- The next step is to apply the element-wise AND (&) operation between the two matrices $[\tau^1]$ and $[\tau^2]$. This results in a third matrix of order $T \times M$, say $[\varphi]$ which is also a binary matrix. The element wise AND (&) operation between $[\tau^1]$ and $[\tau^2]$ is done as:

$$\varphi_{ij} = (\tau_{ij}^{1} \,\&\, \tau_{ij}^{2}) = \begin{cases} 1 & \tau_{ij}^{1} = \tau_{ij}^{2} = 1 \\ 0 & otherwise \end{cases} \tag{9}$$

- In the third step, the algorithm creates another vector $\{\omega\}$ of size $T \times 1$. In this step, all the rows of the $[\varphi]$ matrix are checked, and a value of 1 is stored in the vector $\{\omega\}$ if all the columns of that particular row have values equal to 1. Else, if the value in at least one of the rows in the matrix $[\varphi]$ is 0, the process assigns a value of 0 to the matrix. The binary assignment is shown as follows:

$$\omega_i = \begin{cases} 1 & \varphi_{ij} = 1 \; \forall \; j = 1 \; to \; M \\ 0 & otherwise \end{cases} \tag{10}$$

- The final step involves the assignment of indices of only those rows for which $\{\omega\}$ is equal to 1. Thus, the creation of $\{\theta\}$, the indices vector, can be shown as:

$$\theta_k = i \quad if\ \omega_i = 1 \tag{11}$$

After the above search algorithm runs, the last step is to get the proportion readings of the corresponding combination from the matrix $[\gamma_{T\times N}]$. Hence all the combinations corresponding to the index values stored in $\theta_i{}'s$ are the possible solutions satisfying the desired gradation.

Evaluation of combination closest to the mid-point gradation

Aligning the objective with previous research works, the last step in the model is to choose the optimal option which has the combined gradation closest to the mid-point of the target gradation. Once all the solutions corresponding to indices stored in the vector $\{\theta_{S\times 1}\}$ are known from the $[\gamma_{T\times N}]$ matrix, the sum of the squared deviation for each of the S solutions is calculated and stored in a $\{Z\}$ vector as follows:

$$Z_i = \sum_{k=1}^{k=M} \left[\left(\frac{\beta_{k,1} + \beta_{k,2}}{2}\right) - \gamma_{ik}\right]^2 \tag{12}$$

where, γ_{ik} is read from the $[\gamma_{M\times N}]$ matrix, for only those solutions corresponding to the indices as stored in $\{\theta_S\}$ vector. The solution which has the minimum value of the sum of squared difference as stored in the $\{Z\}$ vector is the one closest to the mid-point gradation and the optimal solution.

Evaluation of combination corresponding to minimum cost

In addition to the above criterion of choosing the best alternative as the one closest to the mid-point gradation, another significant criterion is the cost of the aggregate blending process. This cost may be a combination of quarrying cost, crushing operation and transportation of the aggregates. There may be other factors involving the processing of crushed aggregates which can influence the cost of individual stockpiles of aggregates. Thus, once all the feasible solutions are known and their corresponding indices stored in the vector $\{\theta_{S\times 1}\}$, the proportions are taken from the $[\gamma_{T\times N}]$ matrix and the weighted cost of each of the S solutions is calculated and stored in a $\{C\}$ vector which is defined as follows:

$$C_i = \sum_{k=1}^{k=N} \frac{\gamma_{ik} \times U_k}{100} \tag{13}$$

where γ_{ik} are the proportions of the feasible solutions, and $U_k's$ are the respective unit cost of the respective stockpiles. Thus, the best solution is the one, which has the minimum weighted cost as calculated above, the weights being the unit costs of each stockpile.

An Example

To demonstrate the applicability of the proposed methodology, an example for combining four stockpiles to meet the target gradation is taken. Table 2 shows the sieve size distribution of the individual stockpiles and the required target gradation.

Table 2. Aggregate sieve size distribution data for the given example

Sieve Size (mm) (1)	**Stockpile Designation**								**Specification Limits, as a percentage passing**	
	X_1 (20 mm)		X_2 (10 mm)		X_3 (6.3 mm)		X_4 (2.36 mm)			
	Weight Retained (2)	Percentage Passing (3)	Weight Retained (4)	Percentage Passing (5)	Weight Retained (6)	Percentage Passing (7)	Weight Retained (8)	Percentage Passing (9)	Lower Bound of Target Gradation (10)	Upper Bound of Target Gradation (11)
12.5	209	96.44	0	100.00	0	100.00	0	100.00	90	100
10	4369	22.09	403	88.77	0	100.00	0	100.00	70	88
4.75	1188	1.87	2763	11.79	27	97.52	0	100.00	53	71
2.36	110	0	401	0.61	359	64.59	6	97.60	42	58
1.18	0	0	22	0	446	23.67	74	68.00	34	48
0.6	0	0	0	0	103	14.22	53	46.80	26	38
0.3	0	0	0	0	49	9.72	38	31.60	18	28
0.15	0	0	0	0	22	7.71	22	22.80	12	20
0.075	0	0	0	0	21	5.78	20	14.80	4	10
Pan	0	-	0	-	63	-	37	-	-	-

Unit Cost (U_k) ₹/cubic feet	50	60	30	55	-

Thus, as seen from Table 2, the number of sieves M and the number of stockpiles N are equal to 9 and 4 respectively. The sieve analysis distribution data depicted by columns (2), (4), (6), and (8) of Table 2, is entered by the user as the weight retained in the respective sieves for the four stockpiles, viz., 20, 10, 6.3, and 2.36 mm. The above information entered is used to calculate the percent passing various sieves for each of the stockpiles. The program calculates this information using equation (1) and is shown by the columns (3), (5), (7), and (9) of Table 2. Thus, a matrix $[\alpha_{ij}]$ of order 9×4 is constructed:

$$\alpha_{ij} = \begin{bmatrix} 96.44 & 100.00 & 100.00 & 100.00 \\ \vdots & \vdots & \vdots & \vdots \\ 0.00 & 0.00 & 5.78 & 14.80 \end{bmatrix} \tag{14}$$

The user is now required to enter the lower and upper bounds of the target gradation for each sieve, which is shown by the columns (10) and (11) of Table X. Thus a $[\beta_{ij}]$ matrix of order 9×2 is constructed:

$$\beta_{ij} = \begin{bmatrix} 90 & 100 \\ \vdots & \vdots \\ 4 & 10 \end{bmatrix} \tag{15}$$

The integer solution matrix $[\gamma_{ij}]$, is loaded from the stored database. With N equal to 4, using equation (3) and Table 1, the total number of integer solutions to the summation equation (2), T equals to 176851. Thus, the $[\gamma_{ij}]$ matrix is of the order 176851×4 as shown:

$$\gamma_{ij} = \begin{bmatrix} 0 & 0 & 0 & 100 \\ \vdots & \vdots & \vdots & \vdots \\ 25 & 25 & 25 & 25 \end{bmatrix} \tag{16}$$

The first calculation as explained in the previous section is to calculate the $[\delta_{ij}]$ matrix of the order 176851×9 using equation (6):

$$\delta_{ij} = \begin{bmatrix} 100 & \cdots & 14.80 \\ \vdots & \ddots & \vdots \\ 99.11 & \cdots & 5.14 \end{bmatrix} \tag{17}$$

The next step of the program is to start the search algorithm, as shown above, can be explained using the following sub-steps. The first sub-step is the creation of two matrices, each of order 176851×9, $[\tau^1]$ and $[\tau^2]$ that can be calculated using eq. (7) and (8) as follows:

$$\tau_{ij}^1 = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \tag{18}$$

$$\tau_{ij}^2 = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \tag{19}$$

The next sub-step is to create a $[\varphi]$ matrix by applying element wise AND (&) operation on $[\tau^1]$ and $[\tau^2]$. Thus, a $[\varphi]$ matrix of order 176851×9 is created using equation (9):

$$\varphi_{ij} = (\tau_{ij}^1 \,\&\, \tau_{ij}^2) = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \tag{20}$$

This is followed by the creation of a vector $\{\omega\}$, of order 176851×1 that stores binary values depicting the solutions that satisfy the bound constraints. The $\{\omega\}$ vector is calculated using equation (10) as follows:

$$\omega_i = \begin{Bmatrix} 0 \\ \vdots \\ 0 \end{Bmatrix} \tag{21}$$

The final sub-step is to find and store all the indices of the possible combinations for which ω_i is equal to one. This is stored in a $\{\theta\}$ vector of variable size, which depends on the number of feasible solutions. For instance, in the given example, $\{\theta\}$ is of size 249×1 showing that there are two hundred and forty nine feasible solutions (S) to the problem given. Using equation (11), $\{\theta\}$ is calculated as:

$$\theta = \begin{Bmatrix} 3733 \\ \vdots \\ 94235 \end{Bmatrix} \tag{22}$$

Thus, now we have all the possible solutions that depict the feasible combination of available stockpiles that can be used to reach the target gradation. Some of the feasible solutions are shown in Table 3 and the complete table is given in Appendix A. However, as per the first criterion, the available 249 solutions, the optimal solution is the one that lies closest to the mid-point of the specification limits for the sieves. This is done by calculating the sum of squared deviation for each of the 249 solutions and stored in a $\{Z\}$ vector by using equation (12) as follows:

$$Z_i = \begin{Bmatrix} 8.75 \\ \vdots \\ 8.98 \end{Bmatrix} \tag{23}$$

From the available solutions at our disposal, the combination for which the value of $\{Z\}$ is minimum is summarized in Table 4. As can be seen, proportions for the given stockpiles are $X_1 = 0.37$, $X_2 = 0.06$, $X_3 = 0.00$, $X_4 = 0.57$.

Similarly, the best solution as per the criterion of minimum weighted cost calculated from equation (13), and stored in a C vector as follows:

$$C_i = \begin{Bmatrix} 53.25 \\ \vdots \\ 53.60 \end{Bmatrix} \tag{24}$$

The solution as per this criterion is arrived at a proportion of $X_1 = 0.38$, $X_2 = 0.01$, $X_3 = 0.05$, $X_4 = 0.56$. The same is summarized in Table 5.

Table 3. The feasible solutions for the given example

S. No.	Stockpile Designation				Criterion Vectors	
	X_1 (20 mm)	**X_2 (10 mm)**	**X_3 (6.3 mm)**	**X_4 (2.36 mm)**	**Z**	**C**

1.	38	3	0	59	8.75	53.25
2.	37	4	0	59	8.77	53.35
3.	36	5	0	59	8.79	53.45
4.	37	5	0	58	8.54	53.40
5.	35	6	0	59	8.81	53.55
			⋮			
126.	18	24	1	57	8.82	55.05
127.	24	18	1	57	8.69	54.45
128.	19	22	1	58	9.02	54.90
129.	22	19	1	58	8.96	54.60
130.	19	23	1	57	8.80	54.95
			⋮			
245.	25	14	5	56	8.89	53.20
246.	24	15	5	56	8.91	53.30
247.	23	16	5	56	8.93	53.40
248.	22	17	5	56	8.95	53.50
249.	21	18	5	56	8.98	53.60

Table 4. Optimum Combination lying closest to midpoint target gradation

Blend Characteristics			Blend Gradation		
Characteristics (1)	**Optimum value (2)**	**Specification Requirements (3)**	**Sieve Size (mm) (4)**	**Percentage Passing (5)**	**Specification Limits (6)**
Proportions	X1=0.37	NA	12.5	98.68	100-90
	X2=0.06		10	70.50	88-70
	X3=0.00		4.75	58.40	71-53
	X4=0.57		2.36	55.67	58-42
	8.3163	Minimize	1.18	38.76	48-34

Mean Squared Deviation, as a percentage			0.6	26.68	38-26
			0.3	18.01	28-18
			0.15	13.00	28-12
			0.075	8.44	10-04

Table 5. Best Combination corresponding to the minimum cost

Blend Characteristics			Blend Gradation		
Characteristics (1)	**Optimum value (2)**	**Specification Requirements (3)**	**Sieve Size (mm) (4)**	**Percentage Passing (5)**	**Specification Limits (6)**
Proportions	X1=0.38	NA	12.5	98.65	100-90
	X2=0.01		10	70.28	88-70
	X3=0.05		4.75	61.70	71-53
	X4=0.56		2.36	57.89	58-42
Minimum Cost, in ₹./cubic feet	51.90	Minimize	1.18	39.26	48-34
			0.6	26.92	38-26
			0.3	18.18	28-18
			0.15	13.15	28-12
			0.075	8.58	10-04

Conclusions

A generalized framework is developed for aggregate blending in pavement construction which provides all possible correct proportions for the given stockpiles, proportion closest to the mid-point gradation, and the proportion corresponding to the minimum cost, satisfying all the constraints imposed by any transportation agency. This model is a good substitute to the graphical and trial and error techniques which do not provide all possible solutions and require more time and workforce to achieve the desired goal. The model is implemented by a computer program which gives the solution in almost no time and thereby increases the construction efficiency.

References

Easa, S. M., and Can, E. K. (1985a). “Optimization Model for Aggregate Blending.” *Journal of Construction Engineering and Management*, 111(3), 216–230.

Easa, S. M., and Can, E. K. (1985b). “Stochastic Priority Model for Aggregate Blending.” *Journal of Construction Engineering and Management*, 111(4), 358–373.

Kikuchi, S., Kronprasert, N., and Easa, S. M. (2012). “Aggregate Blending Using Fuzzy Optimization.” *Journal of Construction Engineering and Management*, 138(12), 1411–1420.

Lee, S. M., and Olson, D. L. (1983). “Chance Constrained Aggregate Blending.” *Journal of Construction Engineering and Management*, 109(1), 39–47.

Murthy, V. N. S. (2003). *Geotechnical engineering : principles and practices of soil mechanics and foundation engineering*. Marcel Dekker.

Neumann, D. L. (1964). “Mathematical Method for Blending Aggregates.” *Journal of the Construction Division*, ASCE, 90(2), 1–14.

Ritter, J. B., and Shaffer, L. R. (1962). “Blending Natural Earth Deposits for Least Cost.” *Transactions of the American Society of Civil Engineers*, ASCE, 127(2), 54–77.

Roberts, F. L., Kandhal, P. S., Brown, E. R., Lee, D.-Y., and Kennedy, T. W. (1996). *Hot Mix Asphalt Materials, Mixture Design and Construction*. NAPA Researcha and Education Foundation, Lanham, Maryland.

Toklu, Y. C. (2005). “Aggregate Blending Using Genetic Algorithms.” *Computer-Aided Civil* and *Infrastructure Engineering*, John Wiley & Sons, Ltd (10.1111), 20(6), 450–460.

Appendix A

All 249 feasible solutions for the given example:

S. No.	Stockpile Designation				Criterion Vectors	
	X1 (20 mm)	X2 (10 mm)	X3 (6.3 mm)	X4 (2.36 mm)	Z	C
1	38	3	0	59	8.75	53.25
2	37	4	0	59	8.77	53.35
3	36	5	0	59	8.79	53.45
4	37	5	0	58	8.54	53.40
5	35	6	0	59	8.81	53.55
6	36	6	0	58	8.56	53.50
7	37	6	0	57	8.32	53.45
8	34	7	0	59	8.83	53.65
9	35	7	0	58	8.58	53.60
10	36	7	0	57	8.34	53.55
11	33	8	0	59	8.85	53.75
12	34	8	0	58	8.60	53.70
13	35	8	0	57	8.35	53.65
14	32	9	0	59	8.87	53.85
15	33	9	0	58	8.62	53.80
16	34	9	0	57	8.37	53.75
17	31	10	0	59	8.89	53.95
18	32	10	0	58	8.64	53.90
19	33	10	0	57	8.39	53.85
20	30	11	0	59	8.91	54.05

21	11	31	0	58	9.09	56.00
22	31	11	0	58	8.66	54.00
23	11	32	0	57	8.86	56.05
24	32	11	0	57	8.41	53.95
25	12	29	0	59	9.30	55.85
26	29	12	0	59	8.93	54.15
27	12	30	0	58	9.07	55.90
28	30	12	0	58	8.68	54.10
29	12	31	0	57	8.84	55.95
30	31	12	0	57	8.43	54.05
31	13	28	0	59	9.28	55.75
32	28	13	0	59	8.95	54.25
33	13	29	0	58	9.05	55.80
34	29	13	0	58	8.70	54.20
35	13	30	0	57	8.82	55.85
36	30	13	0	57	8.45	54.15
37	14	27	0	59	9.25	55.65
38	27	14	0	59	8.97	54.35
39	14	28	0	58	9.02	55.70
40	28	14	0	58	8.72	54.30
41	14	29	0	57	8.80	55.75
42	29	14	0	57	8.47	54.25
43	15	26	0	59	9.23	55.55
44	26	15	0	59	8.99	54.45
45	15	27	0	58	9.00	55.60
46	27	15	0	58	8.74	54.40
47	15	28	0	57	8.77	55.65

48	28	15	0	57	8.49	54.35
49	16	25	0	59	9.21	55.45
50	25	16	0	59	9.01	54.55
51	16	26	0	58	8.98	55.50
52	26	16	0	58	8.76	54.50
53	16	27	0	57	8.75	55.55
54	27	16	0	57	8.52	54.45
55	17	24	0	59	9.19	55.35
56	24	17	0	59	9.03	54.65
57	17	25	0	58	8.96	55.40
58	25	17	0	58	8.78	54.60
59	17	26	0	57	8.73	55.45
60	26	17	0	57	8.54	54.55
61	18	23	0	59	9.16	55.25
62	23	18	0	59	9.06	54.75
63	18	24	0	58	8.93	55.30
64	24	18	0	58	8.80	54.70
65	18	25	0	57	8.71	55.35
66	25	18	0	57	8.56	54.65
67	19	22	0	59	9.14	55.15
68	22	19	0	59	9.08	54.85
69	19	23	0	58	8.91	55.20
70	23	19	0	58	8.83	54.80
71	19	24	0	57	8.68	55.25
72	24	19	0	57	8.58	54.75
73	20	21	0	59	9.12	55.05
74	21	20	0	59	9.10	54.95

75	20	22	0	58	8.89	55.10
76	22	20	0	58	8.85	54.90
77	20	23	0	57	8.66	55.15
78	23	20	0	57	8.60	54.85
79	21	21	0	58	8.87	55.00
80	21	22	0	57	8.64	55.05
81	22	21	0	57	8.62	54.95
82	38	3	1	58	8.64	53.00
83	37	4	1	58	8.65	53.10
84	36	5	1	58	8.67	53.20
85	37	5	1	57	8.43	53.15
86	38	1	5	56	8.63	51.90
87	35	6	1	58	8.69	53.30
88	36	6	1	57	8.45	53.25
89	34	7	1	58	8.71	53.40
90	35	7	1	57	8.47	53.35
91	33	8	1	58	8.73	53.50
92	34	8	1	57	8.48	53.45
93	32	9	1	58	8.75	53.60
94	33	9	1	57	8.50	53.55
95	31	10	1	58	8.77	53.70
96	32	10	1	57	8.52	53.65
97	30	11	1	58	8.79	53.80
98	11	31	1	57	8.98	55.75
99	31	11	1	57	8.54	53.75
100	12	29	1	58	9.18	55.60
101	29	12	1	58	8.81	53.90

102	12	30	1	57	8.95	55.65
103	30	12	1	57	8.56	53.85
104	13	28	1	58	9.16	55.50
105	28	13	1	58	8.83	54.00
106	13	29	1	57	8.93	55.55
107	29	13	1	57	8.58	53.95
108	14	27	1	58	9.14	55.40
109	27	14	1	58	8.85	54.10
110	14	28	1	57	8.91	55.45
111	28	14	1	57	8.61	54.05
112	15	26	1	58	9.11	55.30
113	26	15	1	58	8.88	54.20
114	15	27	1	57	8.88	55.35
115	27	15	1	57	8.63	54.15
116	16	25	1	58	9.09	55.20
117	25	16	1	58	8.90	54.30
118	16	26	1	57	8.86	55.25
119	26	16	1	57	8.65	54.25
120	17	24	1	58	9.07	55.10
121	24	17	1	58	8.92	54.40
122	17	25	1	57	8.84	55.15
123	25	17	1	57	8.67	54.35
124	18	23	1	58	9.05	55.00
125	23	18	1	58	8.94	54.50
126	18	24	1	57	8.82	55.05
127	24	18	1	57	8.69	54.45
128	19	22	1	58	9.03	54.90

129	22	19	1	58	8.96	54.60
130	19	23	1	57	8.80	54.95
131	23	19	1	57	8.71	54.55
132	20	21	1	58	9.00	54.80
133	21	20	1	58	8.98	54.70
134	20	22	1	57	8.77	54.85
135	22	20	1	57	8.73	54.65
136	21	21	1	57	8.75	54.75
137	38	2	2	58	8.75	52.70
138	37	3	2	58	8.77	52.80
139	38	2	3	57	8.63	52.45
140	38	3	2	57	8.52	52.75
141	36	4	2	58	8.79	52.90
142	37	4	2	57	8.54	52.85
143	38	2	4	56	8.52	52.20
144	35	5	2	58	8.81	53.00
145	36	5	2	57	8.56	52.95
146	37	2	5	56	8.65	52.00
147	34	6	2	58	8.83	53.10
148	35	6	2	57	8.58	53.05
149	33	7	2	58	8.85	53.20
150	34	7	2	57	8.60	53.15
151	32	8	2	58	8.87	53.30
152	33	8	2	57	8.62	53.25
153	31	9	2	58	8.89	53.40
154	32	9	2	57	8.64	53.35
155	30	10	2	58	8.91	53.50

156	31	10	2	57	8.66	53.45
157	29	11	2	58	8.93	53.60
158	30	11	2	57	8.68	53.55
159	28	12	2	58	8.95	53.70
160	12	29	2	57	9.07	55.35
161	29	12	2	57	8.70	53.65
162	27	13	2	58	8.97	53.80
163	13	28	2	57	9.04	55.25
164	28	13	2	57	8.72	53.75
165	26	14	2	58	8.99	53.90
166	14	27	2	57	9.02	55.15
167	27	14	2	57	8.74	53.85
168	25	15	2	58	9.01	54.00
169	15	26	2	57	9.00	55.05
170	26	15	2	57	8.76	53.95
171	24	16	2	58	9.03	54.10
172	16	25	2	57	8.98	54.95
173	25	16	2	57	8.78	54.05
174	17	24	2	57	8.95	54.85
175	24	17	2	57	8.80	54.15
176	18	23	2	57	8.93	54.75
177	23	18	2	57	8.82	54.25
178	19	22	2	57	8.91	54.65
179	22	19	2	57	8.84	54.35
180	20	21	2	57	8.89	54.55
181	21	20	2	57	8.87	54.45
182	37	3	3	57	8.65	52.55

183	36	4	3	57	8.67	52.65
184	37	3	4	56	8.53	52.30
185	35	5	3	57	8.69	52.75
186	36	3	5	56	8.67	52.10
187	34	6	3	57	8.71	52.85
188	33	7	3	57	8.73	52.95
189	32	8	3	57	8.75	53.05
190	31	9	3	57	8.77	53.15
191	30	10	3	57	8.79	53.25
192	29	11	3	57	8.81	53.35
193	12	28	3	57	9.18	55.05
194	28	12	3	57	8.83	53.45
195	13	27	3	57	9.16	54.95
196	27	13	3	57	8.85	53.55
197	14	26	3	57	9.13	54.85
198	26	14	3	57	8.87	53.65
199	15	25	3	57	9.11	54.75
200	25	15	3	57	8.89	53.75
201	16	24	3	57	9.09	54.65
202	24	16	3	57	8.91	53.85
203	17	23	3	57	9.07	54.55
204	23	17	3	57	8.94	53.95
205	18	22	3	57	9.04	54.45
206	22	18	3	57	8.96	54.05
207	19	21	3	57	9.02	54.35
208	21	19	3	57	8.98	54.15
209	20	20	3	57	9.00	54.25

210	36	4	4	56	8.55	52.40
211	35	4	5	56	8.69	52.20
212	35	5	4	56	8.57	52.50
213	34	6	4	56	8.59	52.60
214	33	7	4	56	8.61	52.70
215	32	8	4	56	8.63	52.80
216	31	9	4	56	8.65	52.90
217	30	10	4	56	8.67	53.00
218	29	11	4	56	8.69	53.10
219	12	28	4	56	9.06	54.80
220	28	12	4	56	8.71	53.20
221	13	27	4	56	9.04	54.70
222	27	13	4	56	8.74	53.30
223	14	26	4	56	9.02	54.60
224	26	14	4	56	8.76	53.40
225	15	25	4	56	9.00	54.50
226	25	15	4	56	8.78	53.50
227	16	24	4	56	8.97	54.40
228	24	16	4	56	8.80	53.60
229	17	23	4	56	8.95	54.30
230	23	17	4	56	8.82	53.70
231	18	22	4	56	8.93	54.20
232	22	18	4	56	8.84	53.80
233	19	21	4	56	8.91	54.10
234	21	19	4	56	8.86	53.90
235	20	20	4	56	8.88	54.00
236	34	5	5	56	8.71	52.30

237	33	6	5	56	8.73	52.40
238	32	7	5	56	8.75	52.50
239	31	8	5	56	8.77	52.60
240	30	9	5	56	8.79	52.70
241	29	10	5	56	8.81	52.80
242	28	11	5	56	8.83	52.90
243	27	12	5	56	8.85	53.00
244	26	13	5	56	8.87	53.10
245	25	14	5	56	8.89	53.20
246	24	15	5	56	8.91	53.30
247	23	16	5	56	8.93	53.40
248	22	17	5	56	8.95	53.50
249	21	18	5	56	8.98	53.60