Proceedings of the INMM & ESARDA Joint Virtual Annual Meeting August 23-26 & August 30-September 1, 2021

Review Of Methods To Aggregate Diversion Detection Probabilities Across Multiple Material Strata

L. Annadevula[1], S. K. Aghara[1], L. Joyce[1], K. Jarman[2], and T. Krieger[3]

[1]University of Massachusetts Lowell, Lowell, MA, USA

[2]Pacific Northwest National Laboratory, Richland, WA, USA

[3]Forschungszentrum Jülich GmbH, Jülich, Germany

ABSTRACT

Statistical models are used to calculate probability of detection of diverted material for individual nuclear material strata to assess the effectiveness of IAEA verification inspections. The safeguards verifications use stratified inventories or materials flows, whereby the material is grouped into strata on the basis of similar physical and chemical characteristics. The detection probabilities (DPs) can be aggregated across material strata to determine the probability of detecting material diversion at the facility level. To aggregate DP to the facility level, one must account for all realistic ways in which a total amount of diverted material may be split among different strata. A brute-force enumeration to calculate aggregate detection probability (ADP), referred to as Partitions method, considers all possible combinations of splitting the goal amount among the existing strata. The algorithm then finds the combination that meets a specified criterion, such as minimum ADP in a facility across all strata. The Partitions method finds the global minimum of the ADP because it explicitly considers all different combinations. However, when considering a large facility with several strata the algorithm comes with a high computational cost. The computational time is observed to increase exponentially with increasing number of strata. Another approach to compute ADP uses Pareto frontiers, inductively updating the frontier by evaluating combinations with one additional stratum at each inductive step. The process iterates until the final stratum is included to calculate the final ADP result. We have studied one other method, which we refer to as a modified Greedy Algorithm. The paper will discuss simulation results for these aggregation algorithms in application to representative mock inspection data and facilities. The proof statements and results presented in this paper complement the work described concurrently in the INMM/ESARDA conference paper by Bevill et al. (2021).

INTRODUCTION

In IAEA nuclear safeguards and inspections, a state is a collection of facilities, a facility is a collection of strata and a stratum is a collection of items and batches. Diversion strategies at stratum level involve different ways of acquiring the goal material from the items or batches and detection strategies involve different selection and identification procedures the inspectors perform to identify the "defected" items (i.e., items with an amount of material removed or added as part of a diversion strategy). The Defect detection probability (DP) is considered as the effectiveness metric in quantifying the effectiveness of various detection strategies on the stratum which was put through specific limiting case diversion strategies [1]. Detection of at least one defect requires selection of the defected items from collection of items followed by identification of at least one defected items as defective using a measurement instrument. Hence DP constitutes Defect selection probability (SP) and Defect identification probability (IP). At facility or state level, the diversion strategies involve different ways of splitting the total goal diversion amount into total number of strata and acquiring the stratum specific goal amounts using the stratum specific diversion strategies which when inspected will result in stratum specific DPs for each stratum which will be aggregated to yield a facility level detection probability (ADP) metric using event independence principle. Among all possible ways or combinations of splitting the total goal amount among existing strata there exists at least one combination which poses maximum proliferation risk yielding (global) minimum aggregate detection probability (Min ADP). Thus, Min ADP is used as limiting case metric to evaluate the combined and overall effectiveness of stratum specific detection strategies at facility or state levels and this paper discusses different techniques and algorithms employed in computation of Min ADP for a discretized problem. Determining the combination of goal amounts that yield minimum of all possible Aggregate Detection Probabilities is considered as an Optimization problem. The stratum level DP curves here, used in estimation of ADPs, are discrete in nature and cannot be represented by any differentiable equation. Therefore, the problem is further sub-classified as combinatorial non-differentiable optimization problem and techniques from heuristics, meta-heuristics and dynamic programming are considered for ADP optimization. From Dynamic Programming point of view the present problem is classified as a one-dimensional allocation problem as material diverted is the only property that requires optimization. That is, all independent variables G_1, G_2, …, G_n of the ADP function defined later in this paper correspond to material diverted from different strata, and 'material diverted or goal amount' is in essence represents a single property.

OVERVIEW OF MOCK FACILITY INFO

Table 1. Mock Enrichment Facility - Stratum Inventory - No. of Items & their respective material compositions in significant quantities (SQ)

Stratum ID	Description	Items	PU (SQ)	HEU (SQ)	LEU (SQ)	NU (SQ)	DU (SQ)	TH (SQ)
UFE	EU 30B PROD CYL IN STORE	26			17.930			
UFN	NU 48Y FEED CYL IN STORE	35				28.967		
UFN-H	NU HEELS CYL IN STORE	40				0.009		
UFD	DU 48Y TAILS CYL IN STORE	300					121.988	
SM1-E	EU SAMPLES AND SLUDGES	163			0.015			
SM1-N	NU SAMPLES AND SLUDGES	34				0.000		
SM1-D	DU SAMPLES AND SLUDGES	42					0.000	
UFEP	EU IN PROCESS CYL	9			8.299			
UFNP	NU IN PROCESS CYL	6				2.298		
UFDP	DU IN PROCESS CYL	2					0.804	

A 10-strata mock enrichment facility example is considered, with each stratum containing different number of item/batches of radioactive materials like plutonium (PU), highly enriched uranium (HEU), low enriched uranium (LEU), natural uranium (NU), depleted uranium (DU), thorium (TH) with different material quantities as shown in table 1. The mock strata are subjected to a diversion strategy where a goal amount of material G is diverted by taking the material from all the items within a stratum equally, yielding the same type of "defect" in each item. This type of diversion strategy makes it harder for the instruments to detect the defects and shall be termed 'Equal' or 'Minimizing IP' diversion strategy. Various detection strategies are employed on each of the defected strata using various instrument types with different number of measurements per instrument type, seal checks and surveillance. Table-2 depicts the type of detection strategies applied on each of the ten strata with exact numbers of instrumental measurements and instrumental RSDs. The C/S correspond to whether items are sealed or under surveillance. Verification I or B correspond to whether verification is done based on items or batches. All the strata here use item-based verifications which means there are no batches in the facility. The rest of the columns given information on number of items measured using respective instrument types and their respective instrumental relative standard deviations (RSDs). The H measurements correspond to gross measurements with 15% RSD, F measurements correspond to partial measurements with 5-10 % RSD, and D measurements correspond to bias measurements with 0.5-1 % RSD. Notional values are depicted in the table 2. The stratum DP curves are evaluated by varying the goal diversion amount from 0 to 2 SQ in steps of 0.01 SQ by applying the said diversion and detection strategies. Once the stratum level DP curves are evaluated using the deterministic or stochastic models described in reference C.Gazze et al [1], a facility-wide diversion strategy is applied where the facility-wide goal amount is varied from 0 to 2 SQ in steps of 0.01 SQ. At each diversion step there will exist many possible combinations of splitting the facility-wide goal amount and acquiring them from the existing 10 strata. These combinations of goal amounts when inputted in stratum level DP curves yield respective stratum level DP values which when aggregated using event independence principle will yield multiple aggregate detection probabilities (ADPs). Computing the worst-case ADP value and its combination of stratum goal amounts provides a measure of the effectiveness of stratum-level detection strategies at the facility-level. The optimization models are built and applied on the mock facility subjected to mock diversion and detection strategies to obtain the combination of material diversions that yield minimum aggregate detection probability.

Table 2. Mock Enrichment Facility - Stratum-level Detection Strategies

Stratum ID	Description	C/S	Verification I or B	H	F	D	RSD H	RSD F	RSD D
UFE	EU 30B PROD CYL IN STORE	N	I	1	2	1	15%	5%	0.50%
UFN	NU 48Y FEED CYL IN STORE	N	I		3	1	15%	7%	0.50%
UFN-H	NU HEELS CYL IN STORE	N	I	1			15%		
UFD	DU 48Y TAILS CYL IN STORE	N	I	6	2	4	15%	9%	1.00%
SM1-E	EU SAMPLES AND SLUDGES	N	I	1			15%		
SM1-N	NU SAMPLES AND SLUDGES	N	I	1			15%		
SM1-D	DU SAMPLES AND SLUDGES	N	I	1			15%		
UFEP	EU IN PROCESS CYL	N	I			1			0.50%
UFNP	NU IN PROCESS CYL	N	I			1			0.50%
UFDP	DU IN PROCESS CYL	N	I			1			1.00%

EXPLORED METHODOLOGIES AND THEIR WORKINGS

Among the explored methods, the brute-force method using algebraic partitions or compositions is taken as the gold standard for validating other methods, because the method will always yield the minimum ADP value and its material combination. The brute force model developed using algebraic compositions explores and evaluates all possible combinations of goal amounts, computes respective ADP values and identifies the combination that has the minimum of all ADP values. As the model evaluates all possible combinations in order to identify the minimum ADP combination, the result thus acquired is always accurate. The process of evaluating all possible combinations and their ADPs becomes computationally expensive both in RAM resources and in execution times. Because the total number of combinations that the model has to evaluate and search for the minimum ADP combination increase nonlinearly with increase in number of strata and total goal diversion amount and increases nonlinearly with decrease in minimum increment size resulting in increase in RAM resources and execution times. To be able to solve the apparent optimization problem using the limited computational power available in the general-purpose computers two viable research options have been explored. The first one involves the development of heuristic optimization techniques that will try to converge to the minimum ADP combination with faster execution times but might be inaccurate. Greedy Algorithm, an example heuristic method, instead of going through all possible combinations like the brute force method employs "stepping-agents", each agent explores the input space in an incremental fashion using the previous step information (calculated minimum ADP combination at a lower goal amount), along its assigned minimum ADP pattern using formulas/checks and steps towards and until they converge at their respective local minimum. The agent with least local minimum combination is taken as minimum ADP combination at the next higher incremental goal amount. This method requires a continuing process of reviewing the results from another more accurate method like brute force to identify patterns of discontinuities. The more such patterns can be recognized and incorporated into the Greedy Algorithm, the more likely results will be accurate. The main advantage of the Greedy Algorithm is that computational time is independent of number of strata, which means it takes almost the same amount of time to evaluate minimum ADP curves for 10 strata and 100 strata mock facilities. A second research option involves development of accurate methods using the principles of dynamic programming. In dynamic programming the overall complex optimization problem is split into multiple simple optimization problems and solved separately and recursively updated to acquire the final result. Splitting into simple problems allows the models to work within the framework of generally available computational resources and acquire results with relatively faster execution times. The pareto frontiers model described by Bevill et al [2,3] is one such method which uses a single pareto frontier per stage and evaluates one stratum with the current frontier in every stage, thereby recursively updating the frontier of global minimum combinations in every stage until combination with the final stratum is evaluated in the final stage. The pareto frontier thus obtained in the final stage contains the combinations and results of the overall global minimum of the original problem. The methodology and development of this dynamic programming approach is discussed by Bevill et al [2]. The present paper provides a proof of optimality of the method. An extension of the Pareto frontiers method that uses multiple parallel pareto frontiers per stage is also discussed here. Results and a proof of global minimum for this extension are also provided. The advantage of this use of multiple parallel frontiers is that it can be parallelized and thereby reduce the execution time and provide the same accuracy as the brute-force and "single" frontier methods.

MODEL PROOFS, RESULTS & VALIDATION WITH BRUTE FORCE METHOD

In this section, possible proof statements of each developed method are presented along with their results presented and validated to that of the brute-force method. Let G be the total goal amount the diverter is planning to acquire in total from all the strata in a facility such that $G_1, G_2, G_3, \ldots, G_n$, represent the amount taken from each individual stratum and $\sum G_i = G$. The function $DP_i(G_i)$ gives the DP curve for the i^{th} stratum of a facility or state as a function of the stratum goal amount G_i. Such DP curves of each stratum are evaluated beforehand and are sampled from in order to compute aggregate detection probabilities (ADPs) as below.

$$Aggregate\ Detection\ Probability\ \boldsymbol{ADP(G)} = 1 - \prod_{i=1}^{n} (1 - DP_i(G_i))$$

Let $C_j = \{G_1, G_2, G_3, \ldots, G_n\}_j$ represent the j^{th} combination of stratum goal amounts which sum up to the total goal amount G and j takes values from 1 to m (some finite number of combinations) and the number of possible combinations m is determined by the total goal amount G, the total number of strata, and the minimum goal increment size. The minimum ADP is given as follows:

$$Minimum\ Aggregate\ Detection\ Probability\ \boldsymbol{Min\ ADP(G)} = \min_{j=1\ to\ m} \left(1 - \prod_{i=1}^{n}\left(1 - DP_i(G_i)\right)\right)_{C_j}$$

Brute Force Model: Proof by definition & Results

The computational model of the brute force method makes use of an algebraic compositions function [4] that uses stars and bars graphical techniques [5] to evaluate all possible ways or combinations C_j of splitting the original goal amount G. Therefore, by definition "the minimum of all possible values of any finite sample space is the global minimum of that sample space", the brute force method always yields the global minimum combination for any specified goal diversion amount G and the brute force results as shown in figure 1(a) are taken as golden standard in validating the results of other methods for calculating or approximating Min ADP.

Greedy Algorithm: Component proof statements & Results

The Greedy Algorithm contains multiple checks or formulas that identify patterns and paths that each of agents uses in order to step through until a local minimum is achieved. The agent whose local minimum has least value compared to that of other agents is taken as Min ADP. In order to identify new patterns, the algorithm makes use of accurate Min ADP results from methods like brute force method or pareto frontiers. Table 3 gives various patterns that are identified from the results of brute force method and table 4 gives the descriptions and component proofs or checks that guide respective agents of the algorithm to step through the input parameter space. The figure 1(b) depicts the Min ADP results from Greedy Algorithm and the results agree with brute-force results of figure 1.

Table 3. Various results from Brute force method used for pattern identification.

Goal Amount	Min ADP	Min ADP Combination	Pattern Number	Agent Number
0.01	0.0028696	"0 0 0 0 0 0 0 **0.01** 0 0"	Pattern-01 (Upper bound Pattern)	All (1,2 & 3)
0.02	0.0058002	"0 0 0 0 0 0 0 **0.02** 0 0"		
0.03	0.0067801	"0 **0.03** 0 0 0 0 0 0 0 0"		
0.04	0.0074324	"0 **0.04** 0 0 0 0 0 0 0 0"		
1.21	0.092207	"0 **0.08** 0 **1.13** 0 0 0 0 0 0"	Pattern-02 (Forward Stepping Pattern)	1
1.22	0.093563	"0 **0.08** 0 **1.14** 0 0 0 0 0 0"		
1.23	0.094942	"0 **0.08** 0 **1.15** 0 0 0 0 0 0"		
1.24	0.096325	"0 **0.09** 0 **1.15** 0 0 0 0 0 0"		
1.25	0.097723	"0 **0.09** 0 **1.16** 0 0 0 0 0 0"		
1.15	0.08441	"0 0 0 **1.15** 0 0 0 0 0 0"	Pattern-03 (Re-Distribution Pattern)	2
1.16	0.085739	"0 **0.08** 0 **1.08** 0 0 0 0 0 0"		
0.01	0.0039088	"0 0 0 **0.01** 0 0 0 0 0 0"	Pattern-04 (Re-Combination Pattern)	3
0.02	0.011983	"0 **0.01** 0 **0.01** 0 0 0 0 0 0"		
0.03	0.01343	"0 0 0 **0.03** 0 0 0 0 0 0"		

Table 4. Description and Component proof statements/checks of identified patterns.

Pattern No	Identified Pattern	Description	Component proof statements /checks
1.	Upper Bound Pattern	The collection of points acquired from computing minimum of all stratum DP values at each diversion amount 'G' act as upper bound to Min ADP at that diversion amount 'G'.	$\text{Min ADP}(G) <= \min_{i=1\, to\, n} DP_i(G)$ For 'n' strata facility, 'G' is any goal amount,
2.	Forward Stepping Pattern	Adding material to a stratum which causes minimum rise in ADP is more likely to yield Min ADP.	$\underset{i=1\, to\, n}{argmin}\left[\frac{DP_i(G_i+dG)-DP_i(G_i)}{1-DP_i(G_i)}\right]$ Above statement gives the stratum-i from which taking the increment amount 'dG' shall cause minimal rise in ADP value. 'G_i' is the current material which has already taken from stratum 'i' before the new increment amount 'dG' is taken from it. Since stratum level DP curve is a non-decreasing function of goal diversion amount 'G'.
3.	Re-Distribution Pattern	Splitting the existing material 'G' from stratum in upper bound pattern to another stratum might lower the ADP. The stratum whose cumulative DP is minimum of all other strata up until current 'G' is the one to acquire the material to lower ADP.	$\underset{i=1\, to\, n}{argmin}\left[\sum_{j=0}^{G} DP_i(j)\right]$ Above statement gives the stratum-i from which existing material from Upper bound has to be split and taken from which will cause least rise in DP_i which in turn causes minimal rise in ADP.
4.	Re-Combination Pattern	Instead of applying forward stepping on multiple stratum simply Re-combining the distributed material and acquiring it from stratum of Upper bound pattern might decrease the ADP.	$\min_{i=1\, to\, n} DP_i(G) < ADP(\text{G1, G2 } \ldots, \text{G}_i)$ Let $G_1, G_2 \ldots, G_i$ be current combination of ADP and $G = \sum_{i=1}^{n} G_i$.

Pareto Frontier Methods: Proof statements on Optimality transfer/Bellman's equation & Results

The proof of optimality transfer or Bellman's equations for the pareto frontier methods are discussed below, stating that such methods will always yield global minimum combinations and are as accurate as the brute force method. We consider two versions of the Pareto frontier method. The first, which we refer to as the Single Frontier method, starts by applying the brute force method to the first two strata from the mock facility and then identifying the minimum DP at each unique amount $G_1 + G_2$ for the combination of these strata (the "frontier"). This "Stage-01" is indicated in figure 2. In stage-02, the stage-01 frontier and third stratum are similarly evaluated to obtain new global minima values and the Pareto frontier is updated with new set of global minimum values. Thus, as shown in figure 3, the Pareto frontier is thus inductively updated using one stratum at a time until the overall global minimum or final pareto frontier is evaluated. By applying the first step of combining two strata across multiple pairs of strata the method can be parallelized to reduce real time of the computation. This Multiple Frontiers method, schematically shown in figure 3, starts by evaluating multiple pairs of frontiers in stage-01 by brute force, giving rise to one frontier per stratum pair. Each such frontier contains the global minimum combinations corresponding to the respective pair of strata. In stage-02, the brute force method is applied on each pair of frontiers reducing the number of frontiers by half or ceil(n_1/2). Similarly, the frontiers are recursively evaluated in pairs in each stage until the final frontier is evaluated in final stage and the final frontier contains the global minimum combinations of the original problem. Through parallelization the total time is reduced but the total brute force calculations remain the same. The proof of optimality of both the frontier methods contains the following components:

1. **Sub-Proof 1**: First-stage Pareto frontiers constitute global minimum combinations due to application of brute force method on each stratum pair,
2. **Sub-Proof 2**: At any stage the updated Pareto frontiers always yield combinations corresponding to global minima as long as previous stage frontiers are made of combinations corresponding to global minima,
3. Using 1 & 2, an inductive argument proves that final stage Pareto frontiers contains combinations corresponding to overall global minimum.

Table 5. Proof statements of Pareto frontier methods (Single and Multiple frontier methods).

Single Frontiers Method - Proof Statements	
Methodology	Figure 3
Nomenclature	Here, i correspond to stratum number & i = 1, 2, 3, …, n (Total number of strata) j corresponds to combination number & j = 1, 2, 3, …, m (Total possible combinations) k corresponds to stage number & k = 1, 2, 3, …, n-1 (Total possible stages)
Sub-Proof 1	$PF_1(G) = \min_{j=1\ to\ m}\left(1 - \prod_{i=1}^{2}\left(1 - DP_{i,j}\left(G_{i,j}\right)\right)\right)$ At stage-1, applying brute force method on stratum 1 & 2 to get stage-1 Pareto frontier (PF_1)
Sub-Proof 2	$\boldsymbol{PF_k}(G) = \min_{j=1\ to\ m}\left(1 - (1 - \boldsymbol{PF_{k-1}}(G_{k-1,j}))(1 - DP_{i=k+1}(G_{k+1,j}))\right)$ At any stage-k, applying brute force method on frontier k-1 & stratum k+1 to get stage-k frontier (PF_k)
Transfer of Optimality	Sub-Proof 1 establishes that the combinations and ADP values of the first stage pareto frontier PF_1 are those of global minima. Sub-Proof 2 is the equivalent Bellman's optimality condition [6, 7, 8] of dynamic programming for the current aggregation problem., through a chaining effect, the condition of global minima is transferred from first stage pareto frontier PF_1 to second PF_2, then to third PF_3 and so on up until final stage pareto frontier PF_{n-1}. Therefore, the final stage pareto frontier will always give minimum ADP values at every point for the apparent n-stratum ADP optimization problem, provided we use the brute force method to update the frontier.
Multiple Frontiers method - Proof Statements	
Methodology	Figure 4
Nomenclature	Here, i correspond to stratum number & i = 1, 2, 3, …, n (Total number of strata) j corresponds to combination number & j = 1, 2, 3, …, m (Total possible combinations) k corresponds to stage number & k = 1, 2, 3, …, x (Total possible stages x = ceil($\log_2$(n/2))) ***l*** corresponds to frontier number & l_k = 1, 2, 3, …, y_k (Total possible frontiers $y_k = y_{k-1}/2$ and y_1=int(n/2))
Sub-Proof 1	$PF_{l,k=1}(G) = \min_{j=1\ to\ m}\left(1 - \prod_{i=2l-1}^{2l}\left(1 - DP_i\left(G_{i,j}\right)\right)\right)$ At stage-1, applying brute force method on stratum pairs 2*l* & 2*l*-1 to get stage-1 Pareto frontier-*l* ($PF_{l,k=1}$)

Sub-Proof 2	$$\boldsymbol{PF_{l,k}}(G) = \min_{j=1\ to\ m}\left(1-\left(1-\boldsymbol{PF_{2l-1,k-1}}\left(G_{k-1,j}\right)\right)\left(1-\boldsymbol{PF_{2l,k-1}}\left(G_{k-1,j}\right)\right)\right)$$ At any stage-k, applying brute force method on frontier pairs 2*l* & 2*l*-1 of previous stage k-1 to get stage-k's frontier-*l* ($PF_{l,k}$)
Transfer of Optimality	The Sub-Proof 1 establishes that the combinations and ADP values of first stage pareto frontier $PF_{l,k=1}$ are that of global minima. Sub-Proof 2 is the equivalent Bellman's optimality condition [6, 7, 8] of dynamic programming for the current aggregation problem., through chaining effect, the condition of global minima is transferred from first stage pareto frontiers $PF_{l,k=1}$ to second $PF_{l,k=2}$, then to third $PF_{l,k=3}$ and so on up until final stage pareto frontiers $PF_{l,k=x}$, where x = ceil(log2(n/2)) is final stage of the approach. Therefore, the final stage pareto frontier will always give minimum ADP values at each and every point for the apparent n stratum ADP optimization problem, provided we use the brute force method of to update the frontier pairs.

The figures 1(c) and 1(d) validates the minimum ADP results from Single and Multiple frontier method with that of brute-force results of figure 1(a).

COMPARISON OF EXECUTION TIMES AMONG ALL METHODS

The execution times (wall times) are evaluated for various models to compare the performance among the models with increase in the number of strata. The same 10-strata enrichment facility data has been repeated 10 times to yield 100-strata facility data. The DP curves are generated for the 100 strata from 0 to 2 SQ with 0.01 SQ as minimum increment size. Then Min ADP curves are evaluated for the generated DP curves by considering first two, first three and so on until first 100 strata using Greedy Algorithm, Single Pareto Frontier method, Multiple Pareto Frontier methods (using loops, using arrays, using dask arrays). Respective execution times are computed simultaneously and are plotted against the number of strata as shown in figure 4. All models except Greedy Algorithm show linear dependence of execution time on number of strata. The Greedy Algorithm has no dependence on the number of strata and always executes around 4 seconds of wall time. All the models are executed on 2.40GHz, 6 Cores, 12 Logical Processors, 16 GB Physical Memory (RAM) computer and the wall times correspond to said computer and may change depending on the computer employed.

CONCLUSIONS

The brute-force method though accurate becomes impractical with increase in number of strata or further refinement of grid. Both these actions result in non-linear increase in total number of possible combinations of acquiring the goal amount from all strata making the brute force method impractical. Based on Bellman's optimality condition, both the single and multiple pareto frontier methods are as accurate as the brute-force method and they come with the computational advantage of being executable on general purpose computers. All frontier models have approximately linear time dependance on number of strata whereas the multiple frontiers model (single-core parallelized) is fastest among all the frontier models. Greedy algorithm being heuristic cannot be as accurate as brute-force method, but the model is exceptionally faster and the execution time is independent of the number of strata.

RECOMMENDATIONS AND FUTURE WORK

The future research involves improving Greedy algorithm's accuracy by further identifying new patterns. The entire process of identifying patterns, building checks and updating Greedy algorithm is currently done manually, a viable way of automating this entire process shall be explored. Also, potential hybrid models by combining Greedy algorithm with frontier models shall be explored.

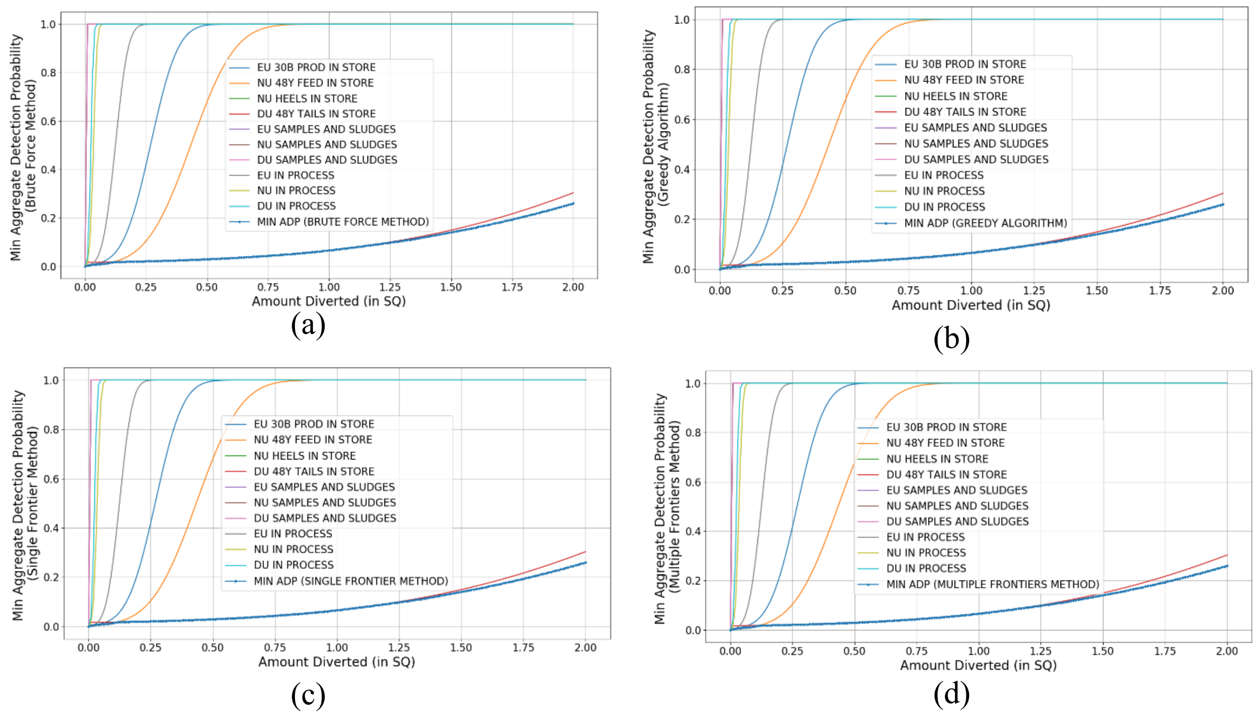

Figure 1. Mock Enrichment Facility stratum DP curves along with minimum ADP curve using (a) Brute Force method (b) Greedy algorithm (c) Single Frontier method and (d) Multiple Frontiers method.

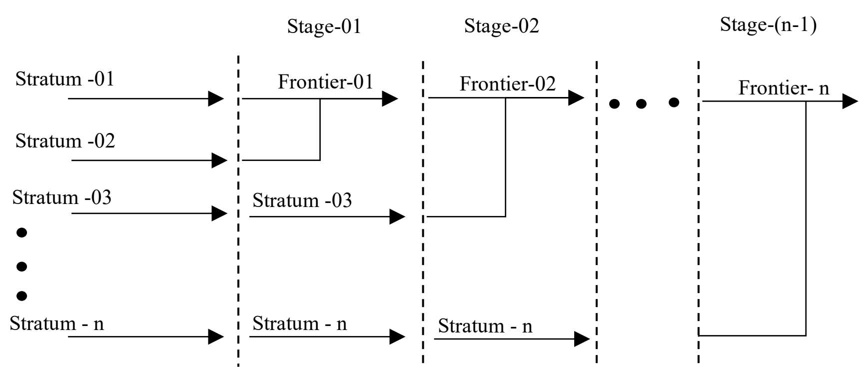

Figure 2. Pareto Frontiers Methodology - Single Frontier Method

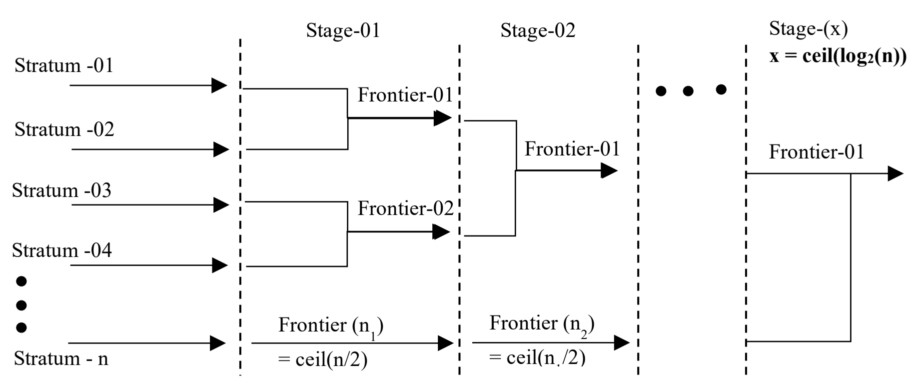

Figure 3. Pareto Frontiers Methodology - Multiple Frontiers Method

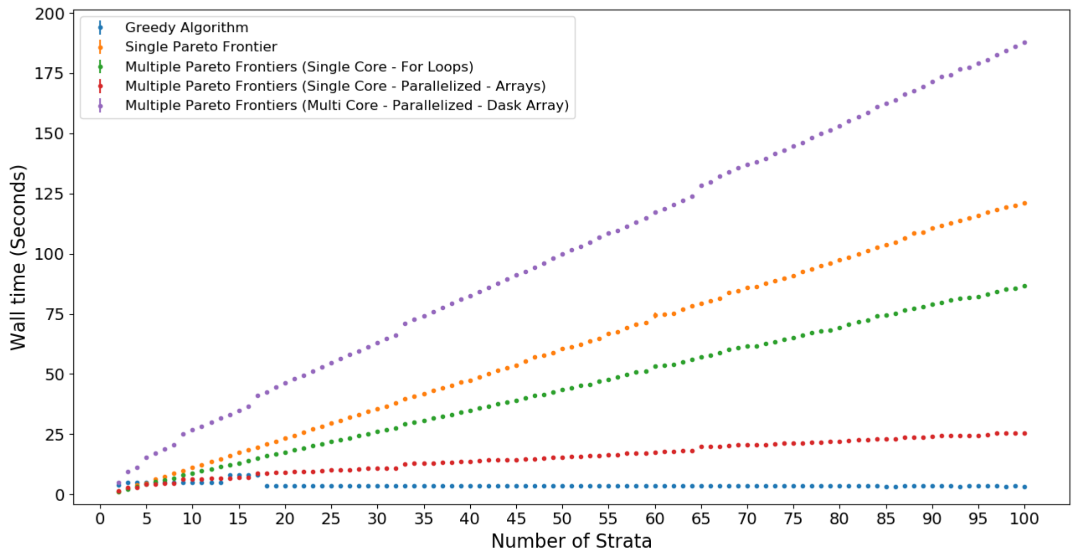

Figure 4. Execution times (wall time) vs. Number of strata for various methods

REFERENCES

1. C. Gazze et al., "Stochastic Approaches for Calculating and Aggregating Detection Probabilities for Nuclear Material Diversion," BNL-211980-2019-INRE, 2019.
2. A. Bevill et al., “Multi-stratum Detection Probability Calculations for IAEA Safeguards: Foundations and Early Progress,” submitted concurrently to Proc. of the INMM & ESARDA Joint Annual Meeting, 2021.
3. A. Bevill, "Detection Probability Estimation for Multi-Stratum Diversion Scenarios," SGIM-IFC Evaluation Libraries, 2019.
4. Python Module, ‘Combalg-py’, https://pythonhosted.org/combalg-py/
5. N. Albert and H. S. Wilf, “Combinatorial Algorithms for Computers and Calculators,” New York: Academic Press, 1978.
6. F. S. Hillier and G. J. Lieberman. “Introduction to operations research,” 11th ed. McGraw Hill 2021. ISBN10: 1259872998 ISBN13: 9781259872990
7. R. E. Bellman, “Dynamic Programming,” Princeton Univ. Press, Princeton, N. J., 1957.
8. R. E. Bellman. S. E. Dreyfus, "One-Dimensional Allocation Processes," in Applied Dynamic Programming: Princeton Legacy Library, 1962, ch. I, pp. 3-38.