STAT906 - Rare Event Simulation with Heavy Tail

Ambrose Emmett-Iwaniw

Department of Statistics and Actuarial Science

UNIVERSITY OF WATERLOO

The Problem

Problem: Ways to estimate $G(x) = \Pr(S_n > x)$ by simulation
where S_n is the sum of n positive i.i.d. r.v.s $X_1, \cdots X_n$ that are Heavy-tailed
In the special case of when x is large which implies G(x) is small.

Applications: Telecommunication, Insurance Risk, Financial Mathematics, etc.

What The Main Paper Covered

Solved the problem using methods that were concerned with using VRT of CMC and IS in the case where n is replaced by N a positive r.v. independent of the Y_i's.

For numerical study of M/G/1 queue where n is replaced by a geometric r.v. N with the X_i's following either Weibull or Pareto distributions.

Proved polynomial time estimators under Weibull or Pareto distributions.

Explored further improvements in CMC in random N case by incorporating CV and Stratication VRTs

Result: The CMC conditioning on $X_1, \cdots X_{n-1}$ is the best estimator

What I will Do Problem Wise: Compound Poisson Process

The numerical experiments to be performed would be on an example stated in the Paper [1] but wasn't investigated was the Compound Poisson Process:

$$X(t) = \sum_{i=1}^{N(t)} X_i$$

Where X_i's are iid and $N(t)$ is a Poisson Process (i.e. $N(t) \sim \text{Poisson}(\lambda t)$)

Where we want to simulate using X_i's as either Weibull or Pareto distributions:

$$\Pr(X(t) > x)$$

What The Slides Present

1. Naive MC Simulation
2. Antithetic Variates
3. Conditional Monte Carlo - Order Statistics
4. Conditional Monte Carlo - Best Condition
5. Taking CMC combining with Stratification
6. Importance sampling - Most Obvious VRT To Try
7. RQMC - How it compares to MC.

What are the Heavy-Tailed Distributions?

As was described in both the papers [1] and [2] the tail of pareto distribution (a regularly varying distribution) with parameter α is:

$$\bar{F}(x) = (1+x)^{-\alpha}$$

the tail of Weibull distribution with parameter β is:

$$\bar{H}(x) = e^{-x^{\beta}}$$

where $F(x) = 1 - \frac{1}{(1+x)^{\alpha}}$ is the CDF of a Pareto with parameter α
where $H(x) = 1 - e^{-x^{\beta}}$ is the CDF of a Weibull with parameter β

The Simulation Setup For Compound Poisson Process

For the Compound Poisson Process the number of random variables added together (aka jumps) follow a Poisson(λt) and so in this case we can hold $\lambda = 1$ and change t and the value of x. Since the goal is to calculate rare events under a heavy-tailed distribution we can simulate rare events we can take $t = 5, 10, 15$ and $x = 50, 75, 100$. For Pareto will take $\alpha = 1.5, 1.75$ (in the threshold of infinite variance) like in the paper. And for Weibull $\beta = 0.5, 0.75$ Will use $n = 10^5, 10^6$ to see how effect n simulations has on the result.

Naive MC Simulation - Setup Of Simulation

The Naive MC simulation estimator of $G(x)$ for n simulations is:

$$\hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} I(X_i(t) > x)$$

where $X_i(t) = \sum_{j=1}^{N_i(t)} X_{ij}$ and $N_i(t) \sim$Poisson(t) are i.i.d. for $i = 1, \cdots n$ and X_{ij} are i.i.d. either Pareto or Weibull distributed.

n	α	t	x	q0.025	Mean	q0.975	n	α	t	x	q0.025	Mean	q0.975
10^4	1.25	5	50	0.047168	0.0515	0.055832	10^6	1.25	5	50	0.0490779	0.049503	0.0499281
10^4	1.25	5	75	0.0251441	0.0284	0.0316559	10^6	1.25	5	75	0.0273604	0.027682	0.0280036
10^4	1.25	5	100	0.0167897	0.0195	0.0222103	10^6	1.25	5	100	0.0181773	0.018441	0.0187047
10^4	1.25	10	50	0.1167579	0.1232	0.1296421	10^6	1.25	10	50	0.129715	0.130375	0.131035
10^4	1.25	10	75	0.0697405	0.0749	0.0800595	10^6	1.25	10	75	0.068589	0.069086	0.069583
10^4	1.25	10	100	0.0394063	0.0434	0.0473937	10^6	1.25	10	100	0.0438967	0.0443	0.0447033
10^4	1.25	15	50	0.2389423	0.2474	0.2558577	10^6	1.25	10	100	0.2495079	0.250357	0.2512061
10^4	1.25	15	75	0.1150963	0.1215	0.1279037	10^6	1.25	15	75	0.1279529	0.128609	0.1292651
10^4	1.25	15	100	0.0734221	0.0787	0.0839779	10^6	1.25	15	100	0.0786378	0.079167	0.0796962
10^4	1.75	5	50	0.0066111	0.0084	0.0101889	10^6	1.75	5	50	0.0065698	0.00673	0.0068902
10^4	1.75	5	75	0.0016019	0.0026	0.0035981	10^6	1.75	5	75	0.0028977	0.003005	0.0031123
10^4	1.75	5	100	7.4144082×10^{-4}	0.0015	0.0022586	10^6	1.75	5	100	0.0017013	0.001784	0.0018667
10^4	1.75	10	50	0.0136331	0.0161	0.0185669	10^6	1.75	10	50	0.0175835	0.017843	0.0181025
10^4	1.75	10	75	0.0049249	0.0065	0.0080751	10^6	1.75	10	75	0.0071292	0.007296	0.0074628
10^4	1.75	10	100	0.0026783	0.0039	0.0051217	10^6	1.75	10	100	0.0038359	0.003959	0.0040821
10^4	1.75	15	50	0.0322535	0.0359	0.0395465	10^6	1.75	15	50	0.0363275	0.036696	0.0370645
10^4	1.75	15	75	0.0095923	0.0117	0.0138077	10^6	1.75	15	75	0.0131727	0.013398	0.0136233
10^4	1.75	15	100	0.0053658	0.007	0.0086342	10^6	1.75	15	100	0.0067635	0.006926	0.0070885

For Pareto Distribution

Naive MC Simulation - Why VRT are Needed Here?

As was referenced in paper [1] the Naive MC simulation of $G(x)$ becomes ever worse as $G(x)$ becomes smaller making the simulation a problem that needs VRTs to improve efficiency as can be seen below:

Antithetic Variates - First Standard VRT To Try

The AV simulation estimator of $G(x)$ for n simulations is:

$$\hat{\mu}_{ant} = \frac{1}{n/2} \sum_{i=1}^{n/2} \frac{Y_i + \tilde{Y}_i}{2}$$

where $Y_i = I(X_i(t) > x), \tilde{Y}_i = I(\tilde{X}_i(t) > x)$, $X_i(t) = \sum_{j=1}^{N_i(t)} X_{ij}$ and $N_i(t) \sim$Poisson(t) are i.i.d. and X_{ij} are i.i.d. for Pareto distributed. but $\tilde{Y}_i$ are generate by the same uniform r.v. in the form 1 - the uniforms. This VRT can't be used since in the i^{th} case the number of dimensions of a uniform r.v. is random and will be different for Y_i and $\tilde{Y}_i$ so can't neccessarly create that dependency in each of the dimensions, which is an assumption this VRT is based on. So AV is bad in this case.

Conditional MC Using Order Statistics - Setup Of Simulation

The Conditional MC simulation estimator of $G(x)$ for n simulations as describe in (1) is:

$$\hat{\mu}_{CMC} = \frac{1}{n}\sum_{i=1}^{n} \Pr(X_i(t) > x | X_{i,(1)} \cdots , X_{i,(N_i(t)-1)})$$

where $X_i(t) = \sum_{j=1}^{N_i(t)} X_{i,j}$ and $N_i(t)$ ∼Poisson(t) are i.i.d. for $i = 1, \cdots , n$ and $X_{i,j}$ are i.i.d. Pareto distributed. Idea: Generate $N_i(t)$ r.v. form the order statistics and remove the largest r.v. also can write this estimator as follows:

$$\hat{\mu}_{CMC} = \frac{1}{n}\sum_{i=1}^{n} \frac{\bar{F}(\max\{x - X_i(t)^*, X_{i,(N_i(t)-1)}\})}{\bar{F}(X_{i,(N_i(t)-1)})}$$

Where $X_i(t)^* = X_{i,(1)} + \cdots + X_{i,(N_i(t)-1)} = X_i(t) - X_{i,(N_i(t))}$

n	α	t	x	q0.025	Mean	q0.975	n	α	t	x	q0.025	Mean	q0.975
10^4	1.25	5	50	0.0469849	0.0490132	0.0510414	10^6	1.25	5	50	0.0487313	0.0489283	0.0491252
10^4	1.25	5	75	0.0266033	0.0278057	0.0290081	10^6	1.25	5	75	0.0274476	0.0275728	0.027698
10^4	1.25	5	100	0.0181963	0.0191756	0.0201548	10^6	1.25	5	100	0.0183789	0.0184678	0.0185568
10^4	1.25	10	50	0.1277835	0.1318768	0.1359701	10^6	1.25	10	50	0.1299446	0.1303502	0.1307559
10^4	1.25	10	75	0.0673269	0.069981	0.0726351	10^6	1.25	10	75	0.0684261	0.0686843	0.0689426
10^4	1.25	10	100	0.0427429	0.0446552	0.0465675	10^6	1.25	10	100	0.0439692	0.0441535	0.0443379
10^4	1.25	15	50	0.2474562	0.2534459	0.2594356	10^6	1.25	10	100	0.2503044	0.2509019	0.2514995
10^4	1.25	15	75	0.121785	0.125742	0.1296989	10^6	1.25	15	75	0.1284305	0.1288349	0.1292393
10^4	1.25	15	100	0.0794094	0.0824377	0.085466	10^6	1.25	15	100	0.0793548	0.0796451	0.0799353
10^4	1.75	5	50	0.0060935	0.0065449	0.0069963	10^6	1.75	5	50	0.0066065	0.0066524	0.0066982
10^4	1.75	5	75	0.0027859	0.0028986	0.0030113	10^6	1.75	5	75	0.0030059	0.0030282	0.0030505
10^4	1.75	5	100	0.0016036	0.0016668	0.0017301	10^6	1.75	5	100	0.0017537	0.0017676	0.0017814
10^4	1.75	10	50	0.0174051	0.0185938	0.0197825	10^6	1.75	10	50	0.0176101	0.0177169	0.0178238
10^4	1.75	10	75	0.0067615	0.0071776	0.0075938	10^6	1.75	10	75	0.0072267	0.007276	0.0073253
10^4	1.75	10	100	0.0037293	0.0039615	0.0041937	10^6	1.75	10	100	0.0040062	0.0040348	0.0040634
10^4	1.75	15	50	0.0338463	0.0357128	0.0375793	10^6	1.75	15	50	0.0362683	0.0364598	0.0366512
10^4	1.75	15	75	0.0124877	0.0133333	0.0141789	10^6	1.75	15	75	0.0133679	0.013454	0.0135402
10^4	1.75	15	100	0.0068207	0.0073495	0.0078783	10^6	1.75	15	100	0.0069271	0.0069745	0.0070219

For Pareto Distribution, reduction on average of about half in variance, also can be seen near very small probabilities especially $10^4, 1.75, 10, 100$ CI calculated well.

Conditional MC Using Most Efficient - Setup Of Simulation

As given in paper (1) we have the identity:

$$\Pr(X_i(t) > x) = N_i(t) \Pr(X_i(t) > x, X_{i,(N_i(t))} = X_{i,N_i(t)})$$

The Conditional MC simulation most efficient estimator of $G(x)$ for n simulations as described in (1) is:

$$\hat{\mu}_{CMC} = \frac{1}{n} \sum_{i=1}^{n} N_i(t) \Pr(X_i(t) > x, X_{i,(N_i(t))} = X_{i,N_i(t)} | X_1 \cdots X_{N_i(t)-1})$$

where $X_i(t) = \sum_{j=1}^{N_i(t)} X_{i,j}$ and $N_i(t) \sim$Poisson(t) are i.i.d. for $i = 1, \cdots, n$ and $X_{i,j}$ are i.i.d. Pareto distributed. Also can write this estimator as follows:

$$\hat{\mu}_{CMC} = \frac{1}{n} \sum_{i=1}^{n} N_i(t) \bar{F}(\max\{X_{i,(N_i(t)-1)}, x - S_{N_i(t)-1}\})$$

Where $S_{N_i(t)-1} = X_{i,1} + \cdots + X_{i,N_i(t)-1} = X_i(t) - X_{i,N_i(t)}$

n	α	t	x	Mean	HW Ratio	n	α	t	x	Mean	HW Ratio
10000	1.25	5	50	0.04901	6.1569187	1000000	1.25	5	50	0.049009	6.1449275
10000	1.25	5	75	0.027844	8.7444444	1000000	1.25	5	75	0.027591	9.1428571
10000	1.25	5	100	0.018472	11.7031963	1000000	1.25	5	100	0.018458	12
10000	1.25	10	50	0.131133	3.6132537	1000000	1.25	10	50	0.13027	3.5483871
10000	1.25	10	75	0.068335	5.6617312	1000000	1.25	10	75	0.068858	5.6477273
10000	1.25	10	100	0.044069	7.6471735	1000000	1.25	10	100	0.044081	7.9019608
10000	1.25	15	50	0.249671	2.2991155	1000000	1.25	10	100	0.250376	2.2911051
10000	1.25	15	75	0.128555	3.7547387	1000000	1.25	15	75	0.128714	3.7758621
10000	1.25	15	100	0.079319	5.5658436	1000000	1.25	15	100	0.079457	5.4639175
10000	1.75	5	50	0.006646	16.5483871	1000000	1.75	5	50	0.00664	17.7777778
10000	1.75	5	75	0.003048	32.6111111	1000000	1.75	5	75	0.003029	26.75
10000	1.75	5	100	0.001759	48.3157895	1000000	1.75	5	100	0.001761	41
10000	1.75	10	50	0.017785	8.9442379	1000000	1.75	10	50	0.017725	9.5555556
10000	1.75	10	75	0.007256	19.3103448	1000000	1.75	10	75	0.007288	18.5555556
10000	1.75	10	100	0.004054	27.8292683	1000000	1.75	10	100	0.004038	31.25
10000	1.75	15	50	0.036954	5.7136364	1000000	1.75	15	50	0.036347	5.8253968
10000	1.75	15	75	0.013407	12.5428571	1000000	1.75	15	75	0.013421	12.5555556
10000	1.75	15	100	0.007019	19.8888889	1000000	1.75	15	100	0.007014	20.5

For Pareto Distribution, This was efficient but only for the fixed n but in this case it does very well. We will try next stratification to eliminate this.

Conditional MC Using Stratification - Setup Of Simulation

To remove the variability produced by $N(t)$ we will use Stratication with proportional allocation and for $t = 5$ will divide into 8 strata $N_i = i$ for $i = 0, \cdots 7$, and $N_8 > 7$ and $t = 10$ will divide into 15 strata $N_i = i$ for $i = 0, \cdots 14$, and $N_{15} > 15$ and $t = 15$ will divide into 21 strata $N_i = i$ for $i = 0, \cdots 20$, and $N_{21} > 20$

n	α	t	x	q0.025	Mean	q0.975	n	α	t	x	q0.025	Mean	q0.975
10000	1.25	5	50	0.084725	0.089216	0.093707	1000000	1.25	5	50	0.089127	0.08958	0.090033
10000	1.25	5	75	0.050334	0.054088	0.057842	1000000	1.25	5	75	0.0498	0.05015	0.0505
10000	1.25	5	100	0.030535	0.033321	0.036107	1000000	1.25	5	100	0.033152	0.033441	0.03373
10000	1.25	10	50	0.244895	0.252386	0.259877	1000000	1.25	10	50	0.249158	0.249905	0.250652
10000	1.25	10	75	0.122122	0.127548	0.132974	1000000	1.25	10	75	0.131459	0.132032	0.132605
10000	1.25	10	100	0.075904	0.080081	0.084258	1000000	1.25	10	100	0.083855	0.084325	0.084795
10000	1.25	15	50	0.471331	0.481002	0.490673	1000000	1.25	10	100	0.48471	0.4857	0.48669
10000	1.25	15	75	0.249763	0.257613	0.265463	1000000	1.25	15	75	0.249743	0.250506	0.251269
10000	1.25	15	100	0.14747	0.153703	0.159936	1000000	1.25	15	100	0.153998	0.154622	0.155246
10000	1.75	5	50	0.010666	0.012467	0.014268	1000000	1.75	5	50	0.01192	0.012093	0.012266
10000	1.75	5	75	0.004993	0.006447	0.007901	1000000	1.75	5	75	0.005355	0.005474	0.005593
10000	1.75	5	100	0.002234	0.003155	0.004076	1000000	1.75	5	100	0.003141	0.003236	0.003331
10000	1.75	10	50	0.031453	0.034602	0.037751	1000000	1.75	10	50	0.03373	0.034026	0.034322
10000	1.75	10	75	0.0122	0.014164	0.016128	1000000	1.75	10	75	0.013927	0.014129	0.014331
10000	1.75	10	100	0.006222	0.007531	0.00884	1000000	1.75	10	100	0.007533	0.007682	0.007831
10000	1.75	15	50	0.066567	0.070681	0.074795	1000000	1.75	15	50	0.07067	0.071085	0.0715
10000	1.75	15	75	0.023873	0.02638	0.028887	1000000	1.75	15	75	0.026086	0.026356	0.026626
10000	1.75	15	100	0.013106	0.01547	0.017834	1000000	1.75	15	100	0.013531	0.013731	0.013931

For Pareto Distribution, This was seems to have given a better result by smoothing the variance out of the Poisson r.v.

Importance Sampling Using Method in Paper - Setup Of Simulation

As given in paper (1) we have the identity:

$$\Pr(X_i(t) > x) = N_i(t)\Pr(X_i(t) > x, X_{i,(N_i(t))} = X_{i,N_i(t)})$$

The IS simulation estimator of $G(x)$ for n simulations as decribe in (1) is:

$$\hat{\mu}_{IS} = \frac{1}{n}\sum_{i=1}^{n} N_i(t)\frac{f(X_{i,N_i(t)})}{f^*(X_{i,N_i(t)})} I(X_i(t) > x, X_{i,(N_i(t))} = X_{i,N_i(t)})$$

where $f(x) = \frac{\alpha}{(1+x)^{1+\alpha}}$ is density of Pareto with shape α and $f^*(x) = \frac{\alpha^*}{(1+x)^{1+\alpha^*}}$ is density of Pareto with shape $\alpha^* = b/\log(x)$, where b is some cst say $b = 5$ and $X_i(t) = \sum_{j=1}^{N_i(t)} X_{i,j}$ and $N_i(t)$ ~Poisson(t) are i.i.d. for $i = 1, \cdots, n$ and $X_{i,j}$ are i.i.d. Pareto distributed.

n	α	t	x	q0.025	Mean	q0.975	n	α	t	x	q0.025	Mean	q0.975
10000	1.25	5	50	0.037539	0.048762	0.059985	1000000	1.25	5	50	0.054957	0.056179	0.057401
10000	1.25	5	75	0.012816	0.018207	0.023598	1000000	1.25	5	75	0.018325	0.018884	0.019443
10000	1.25	5	100	0.007428	0.011234	0.01504	1000000	1.25	5	100	0.008881	0.009214	0.009547
10000	1.25	10	50	0.116728	0.143188	0.169648	1000000	1.25	10	50	0.139628	0.142241	0.144854
10000	1.25	10	75	0.034998	0.047188	0.059378	1000000	1.25	10	75	0.046255	0.04747	0.048685
10000	1.25	10	100	0.018363	0.025934	0.033505	1000000	1.25	10	100	0.0213	0.022002	0.022704
10000	1.25	15	50	0.234301	0.27831	0.322319	1000000	1.25	10	100	0.265611	0.269919	0.274227
10000	1.25	15	75	0.082597	0.104737	0.126877	1000000	1.25	15	75	0.090151	0.09221	0.094269
10000	1.25	15	100	0.033935	0.046367	0.058799	1000000	1.25	15	100	0.039228	0.040392	0.041556
10000	1.75	5	50	0.000006	0.000554	0.001102	1000000	1.75	5	50	0.001084	0.001159	0.001234
10000	1.75	5	75	−0.000093	0.000221	0.000535	1000000	1.75	5	75	0.000262	0.000288	0.000314
10000	1.75	5	100	−0.000058	0.000079	0.000216	1000000	1.75	5	100	0.000098	0.000111	0.000124
10000	1.75	10	50	0.003337	0.005689	0.008041	1000000	1.75	10	50	0.00354	0.003734	0.003928
10000	1.75	10	75	0.000381	0.001181	0.001981	1000000	1.75	10	75	0.000702	0.000764	0.000826
10000	1.75	10	100	−0.000005	0.00031	0.000625	1000000	1.75	10	100	0.000207	0.000233	0.000259
10000	1.75	15	50	0.007137	0.011587	0.016037	1000000	1.75	15	50	0.007599	0.007962	0.008325
10000	1.75	15	75	0.000543	0.001543	0.002543	1000000	1.75	15	75	0.001321	0.001429	0.001537
10000	1.75	15	100	−0.000094	0.000231	0.000556	1000000	1.75	15	100	0.000385	0.00043	0.000475

For Pareto Distribution, The results seem to be better than Naive MC but not by much. This is a similar problem to CMC before and adding either CV or Stratification can make a big difference in smoothing out the Poisson r.v.

RQMC - Setup Of Simulation

The RQMC simulation estimator of $G(x)$ for n simulations is:

$$\hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} I(X_i(t) > x)$$

where $X_i(t) = \sum_{j=1}^{N_i(t)} X_{ij}$ and $N_i(t)$ ~Poisson(t) are i.i.d. for $i = 1, \cdots n$ and X_{ij} are i.i.d. Pareto distributed but using Sobol with digital-shift to have low-discrepancy in the uniforms used to generate these r.v. to add dependences to lower the variance.

n	α	t	x	q0.025	Mean	q0.975	n	α	t	x	q0.025	Mean	q0.975
10^4	1.25	5	50	0.03998	0.044	0.04802	10^6	1.25	5	50	0.0425745	0.042972	0.0433695
10^4	1.25	5	75	0.0202494	0.0232	0.0261506	10^6	1.25	5	75	0.0244505	0.024755	0.0250595
10^4	1.25	5	100	0.0166966	0.0194	0.0221034	10^6	1.25	5	100	0.0167516	0.017005	0.0172584
10^4	1.25	10	50	0.1031843	0.1093	0.1154157	10^6	1.25	10	50	0.1115894	0.112208	0.1128266
10^4	1.25	10	75	0.0551524	0.0598	0.0644476	10^6	1.25	10	75	0.0578238	0.058283	0.0587422
10^4	1.25	10	100	0.0351101	0.0389	0.0426899	10^6	1.25	10	100	0.0377447	0.03812	0.0384953
10^4	1.25	15	50	0.2143476	0.2225	0.2306524	10^6	1.25	10	100	0.2296815	0.230507	0.2313325
10^4	1.25	15	75	0.098504	0.1045	0.110496	10^6	1.25	15	75	0.107244	0.107852	0.10846
10^4	1.25	15	100	0.0620994	0.067	0.0719006	10^6	1.25	15	100	0.0655592	0.066046	0.0665328
10^4	1.75	5	50	0.0045738	0.0061	0.0076262	10^6	1.75	5	50	0.0057124	0.005862	0.0060116
10^4	1.75	5	75	0.0014409	0.0024	0.0033591	10^6	1.75	5	75	0.0026523	0.002755	0.0028577
10^4	1.75	5	100	5.9374911×10^{-4}	0.0013	0.0020063	10^6	1.75	5	100	0.0015822	0.001662	0.0017418
10^4	1.75	10	50	0.0135406	0.016	0.0184594	10^6	1.75	10	50	0.0139751	0.014207	0.0144389
10^4	1.75	10	75	0.0058089	0.0075	0.0091911	10^6	1.75	10	75	0.0061054	0.00626	0.0064146
10^4	1.75	10	100	0.002093	0.0032	0.004307	10^6	1.75	10	100	0.00346	0.003577	0.003694
10^4	1.75	15	50	0.022222	0.0253	0.028378	10^6	1.75	15	50	0.0267141	0.027032	0.0273499
10^4	1.75	15	75	0.0087741	0.0108	0.0128259	10^6	1.75	15	75	0.0107728	0.010977	0.0111812
10^4	1.75	15	100	0.0039635	0.0054	0.0068365	10^6	1.75	15	100	0.005818	0.005969	0.00612

For Pareto Distribution, Slightly lower variance than Naive MC.

RQMC Simulation - Issues found

The problem with using RQMC here it is very inefficient compared to naive MC most likely because of Sobol sequences being generated are relatively less efficient than sampling straight from the r.v. itself, also couldn't use the m repetitions approach since running time was to long.

What Wasn't Covered

Below is a list of experiments that I will look into in the final paper:

- Look into other Heavy-tail distribution discussed in paper Weibull
- Taking CMC combining with CV
- For Both CV and Stratification combined with efficient estimator add RQMC.
- Investigate other heavy-tailed distributions like lognormal or Levy
- Using weighted IS with loggamma

References

1 Asmussen, S. and Kroese, D (2006) Improved algorithms for rare event simulation with heavy tails. Advances in Appied Probability,38(2), 545-555

2 Asmussen, S., Binswanger, K. and HOjaard, B. (2000). Rare events simulation for heavy-tailed distributions. Bernoulli 6, 303-322

3. Levy Distribution to model price changes

4. Lognormal random walks

5. Random Sampling Pareto r.v.

6. SANDEEP JUNEJA and PERWEZ SHAHABUDDIN. Simulating Heavy Tailed Processes Using Delayed Hazard Rate Twisting