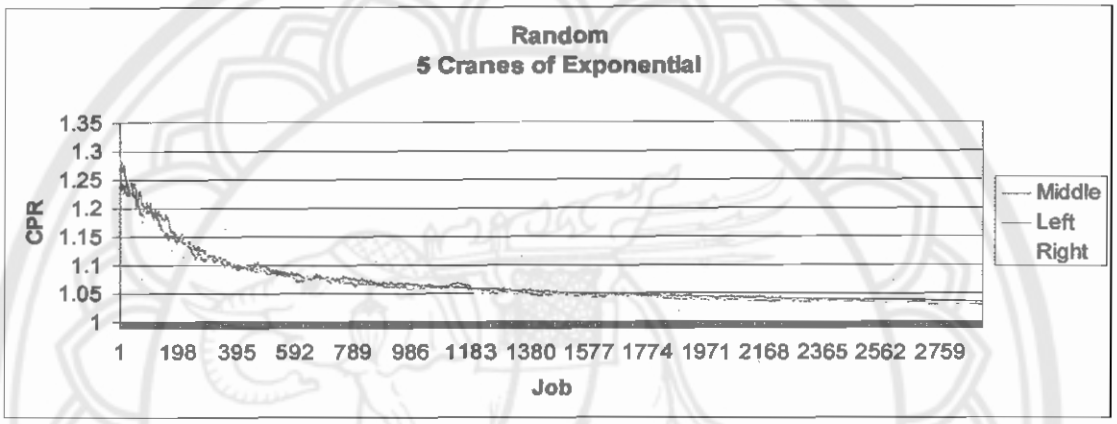
ภาคผนวก ก
กราฟและตารางแสดงผลการทดลอง

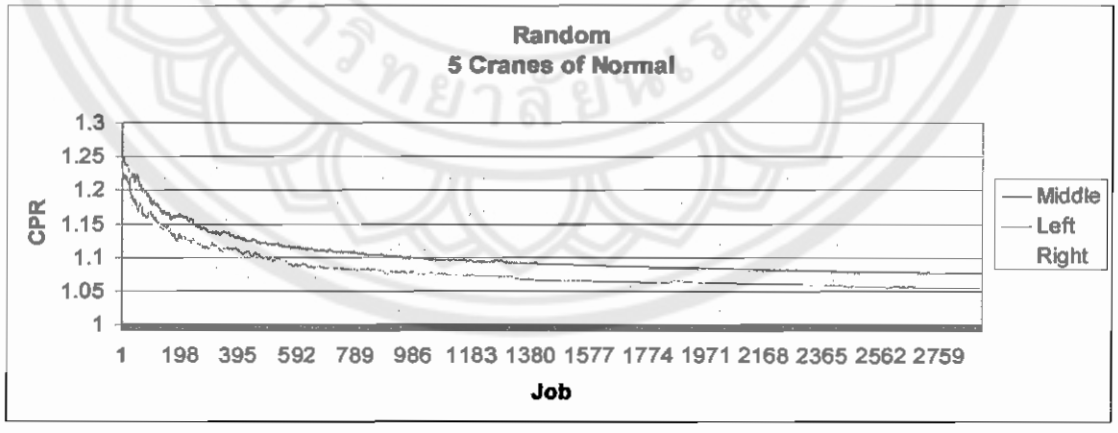
ภาคผนวก ก

กราฟและตารางแสดงผลการทดลอง

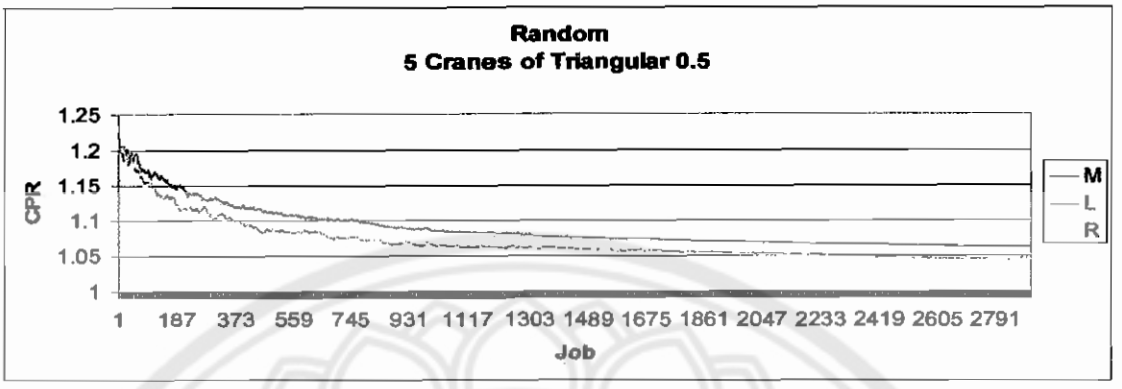
จะเป็นการแสดงกราฟและตารางทางสถิติเปรียบเทียบของวิธีการจัดลำดับการทำงานของเครนวิธี Middle Heuristic, Left Heuristic และ Right Heuristic ของแต่ละการแจกแจงเวลา แต่ละวิธีการสุ่มตำแหน่งงาน และแต่ละจำนวนเครน



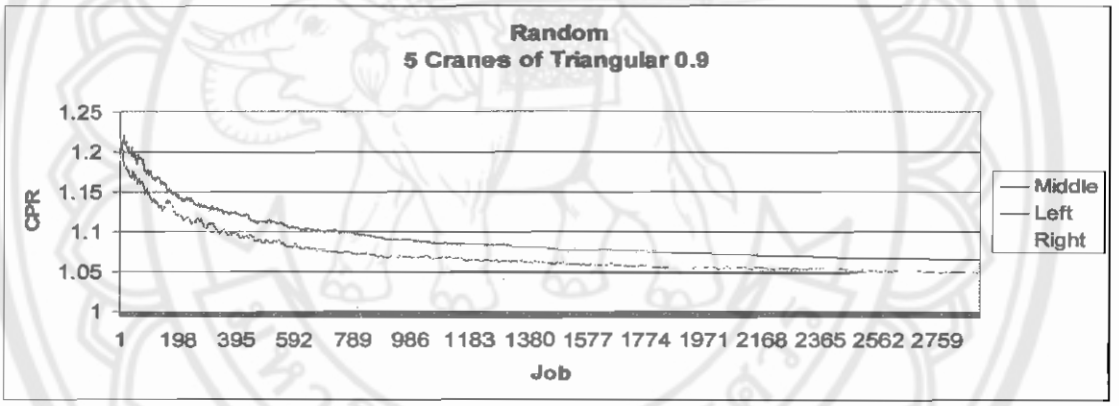
รูปที่ ก-1 แสดงกราฟ การแจกแจงเวลาแบบ Exponential ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ Random



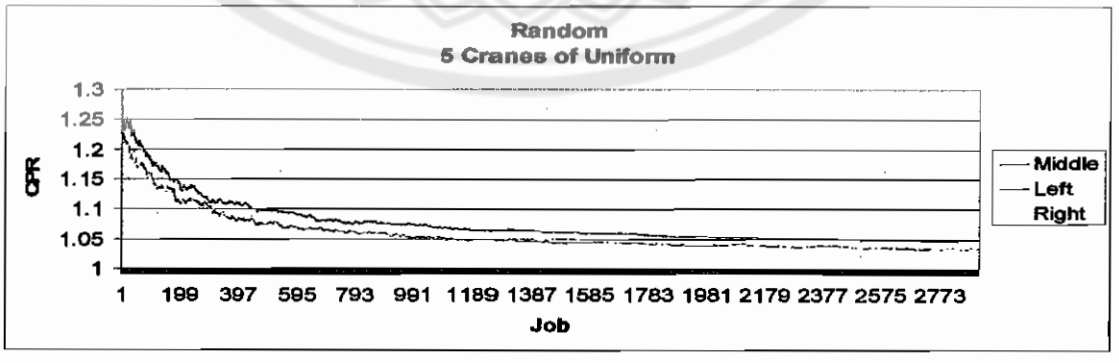
รูปที่ ก-2 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ Random



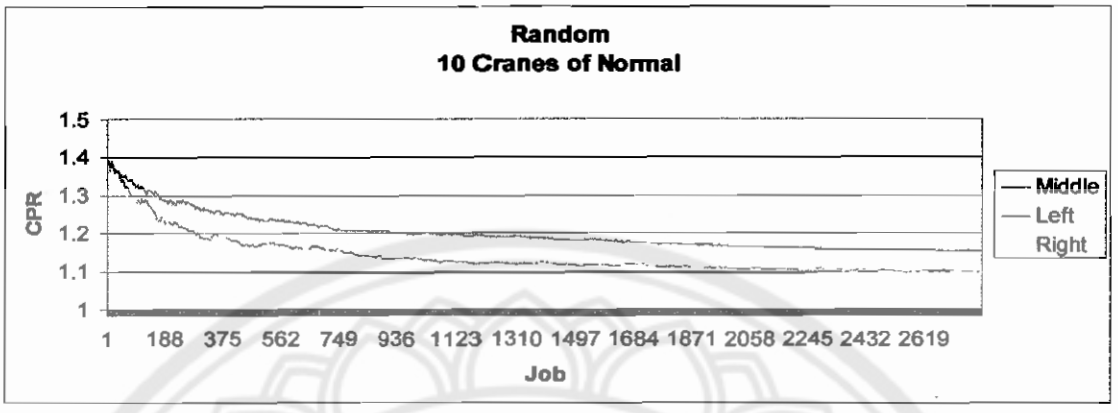
รูปที่ ก-3 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 **ของจำนวนเครน** 5 **เครน และ การสุ่มตำแหน่งแบบ** Random



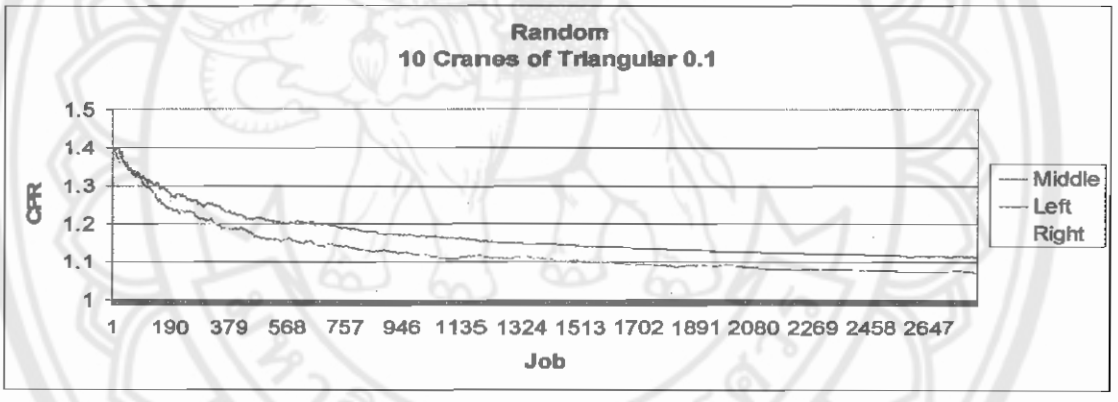
รูปที่ ก-4 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 **ของจำนวนเครน** 5 **เครน และ การสุ่มตำแหน่งแบบ** Random



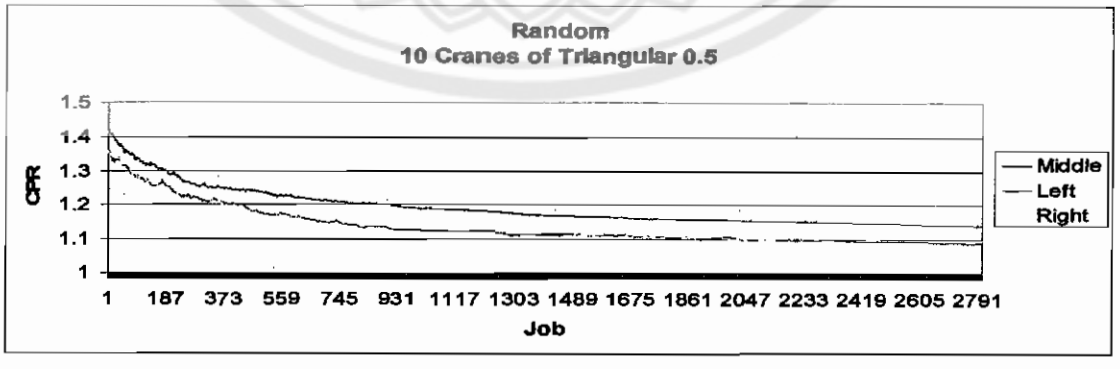
รูปที่ ก-5 แสดงกราฟ การแจกแจงเวลาแบบ Uniform **ของจำนวนเครน** 5 **เครน และการสุ่มตำแหน่งแบบ** Random



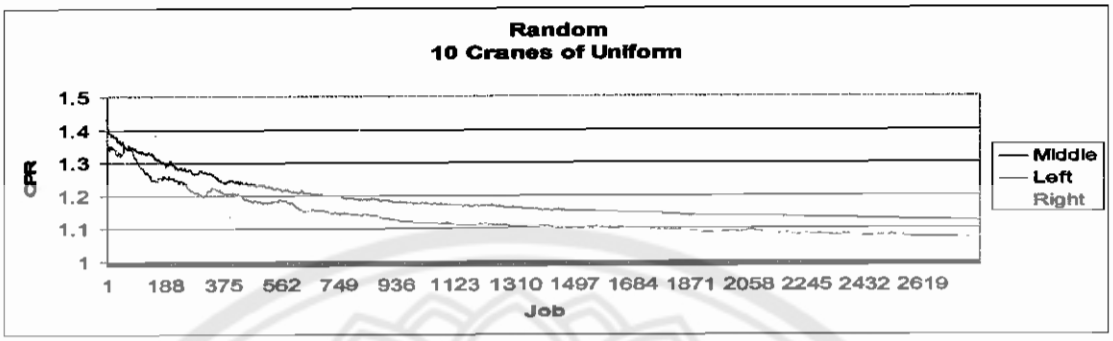
รูปที่ ก-6 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Rando



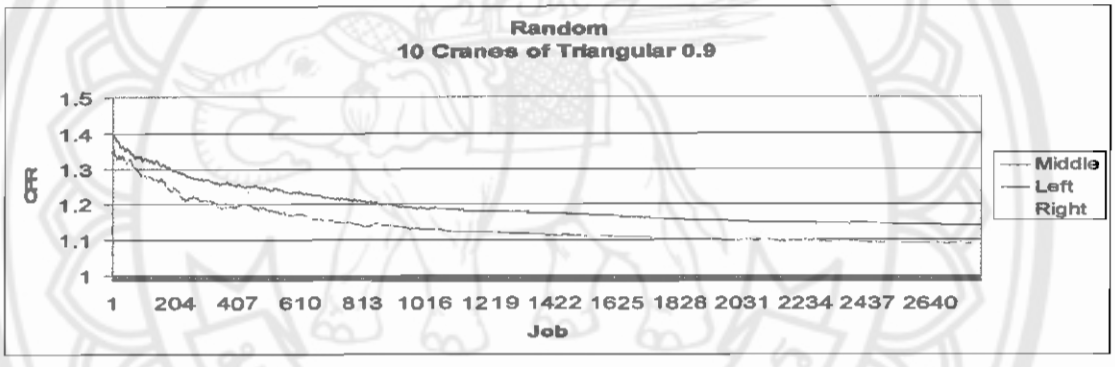
รูปที่ ก-7 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Random



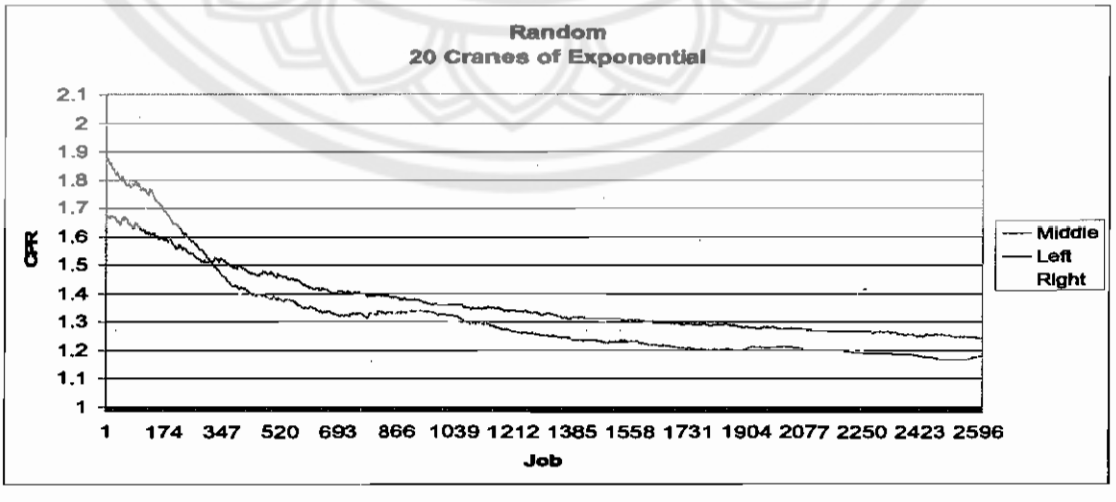
รูปที่ ก-8 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Random



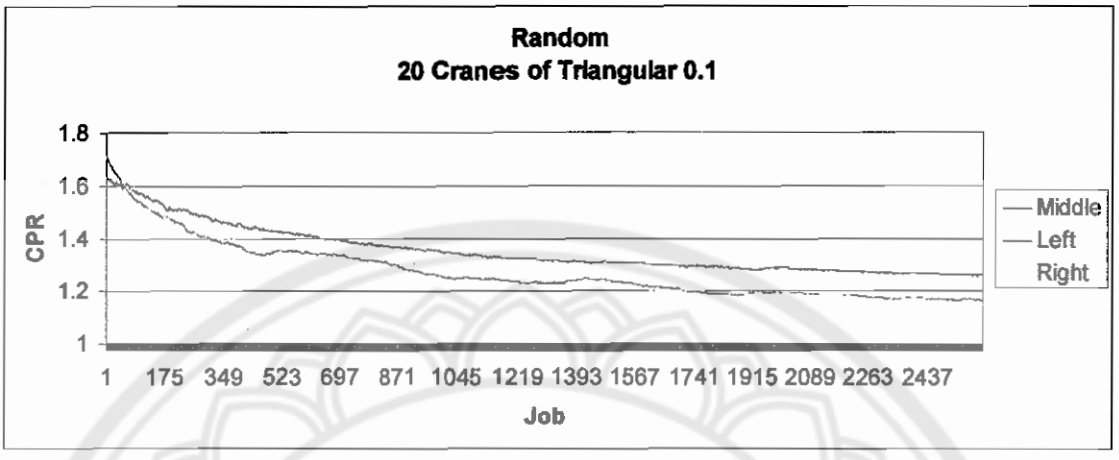
รูปที่ ก-9 แสดงกราฟ การแจกแจงเวลาแบบ Uniform ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Random



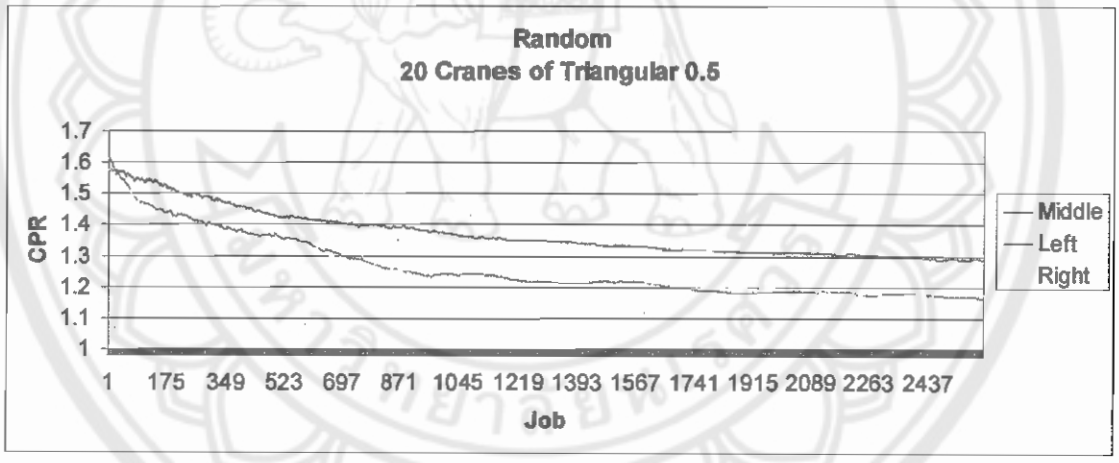
รูปที่ ก-10 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Random



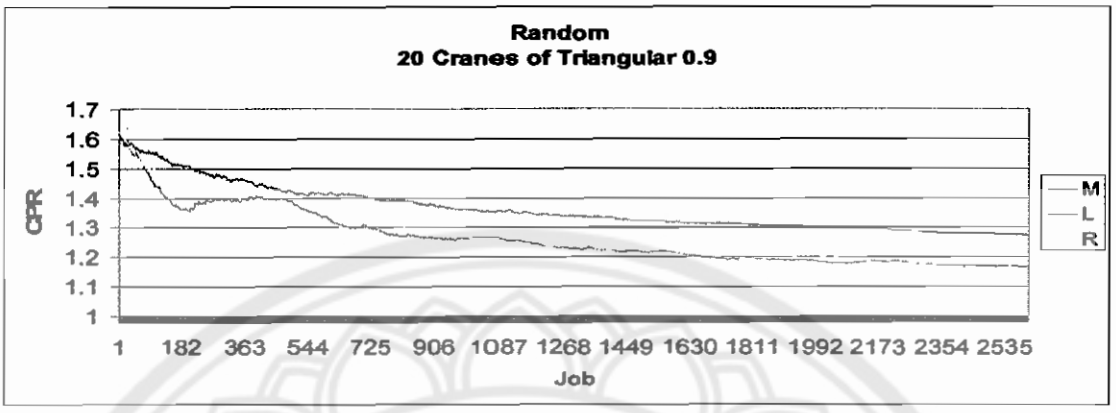
รูปที่ ก-11 แสดงกราฟ การแจกแจงเวลาแบบ Exponential ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Random



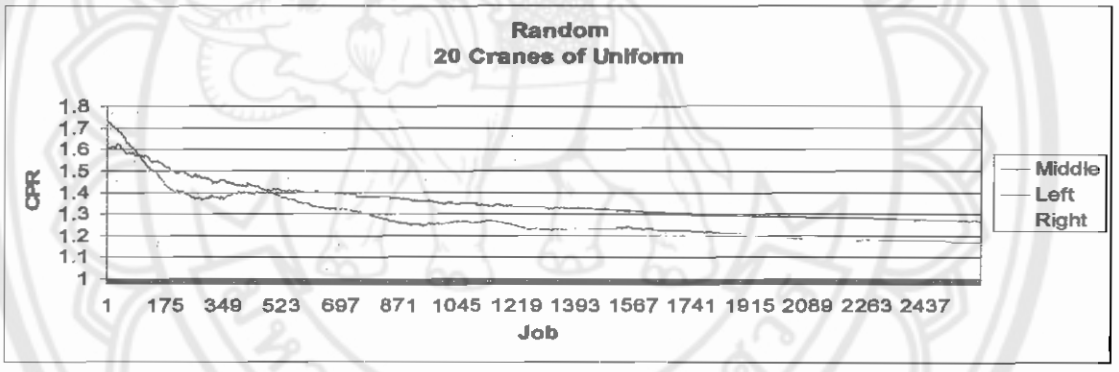
รูปที่ ก-12 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Random



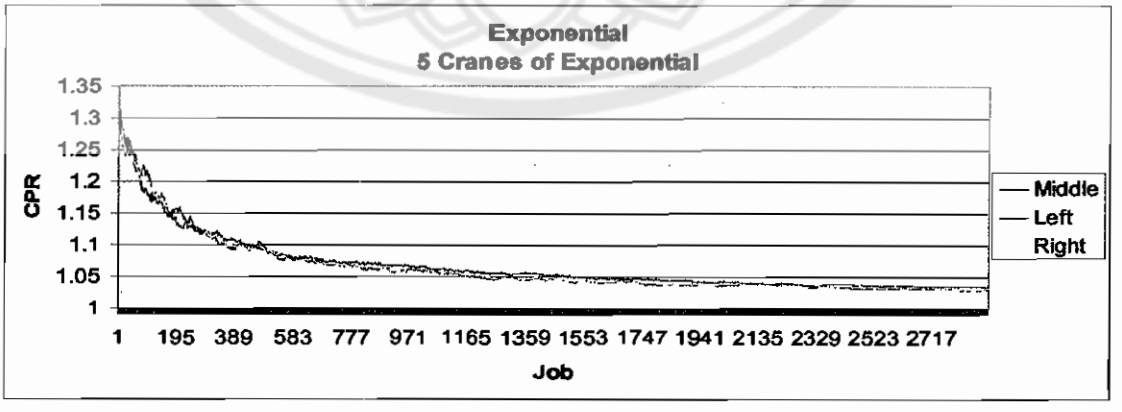
รูปที่ ก-13 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Random



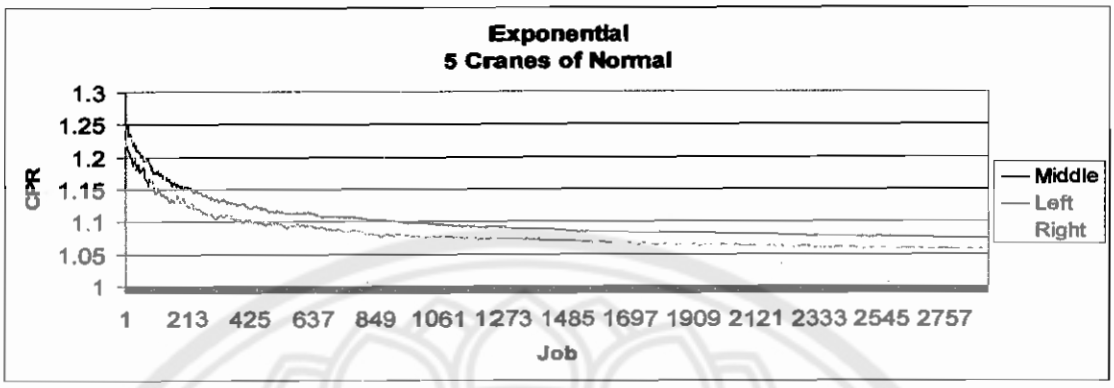
รูปที่ ก-14 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Random



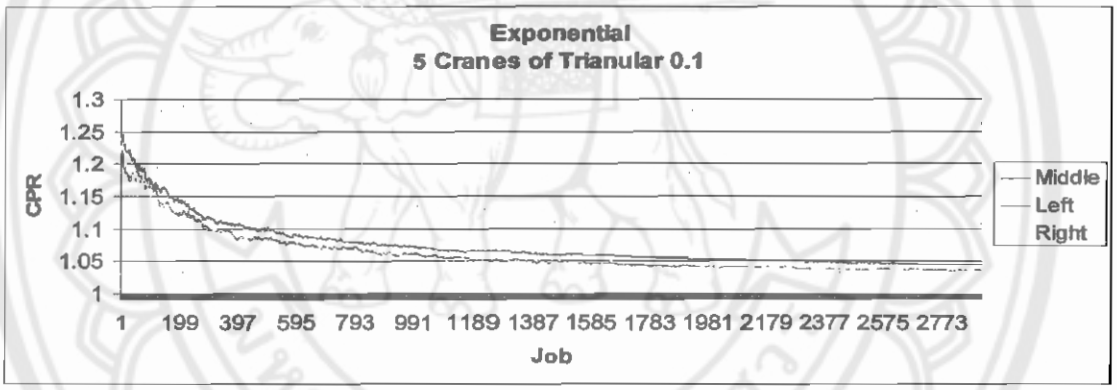
รูปที่ ก-15 แสดงกราฟ การแจกแจงเวลาแบบ Uniform ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Random



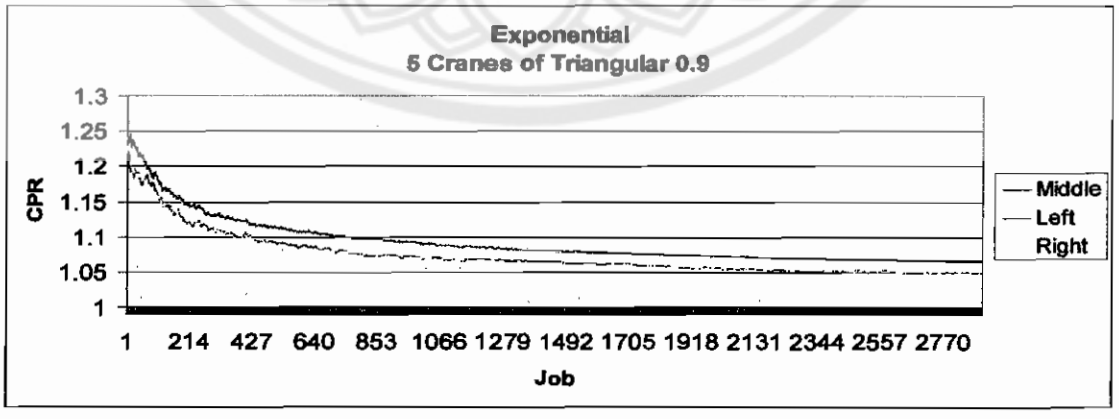
รูปที่ ก-16 แสดงกราฟ การแจกแจงเวลาแบบ Exponential ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ Exponential



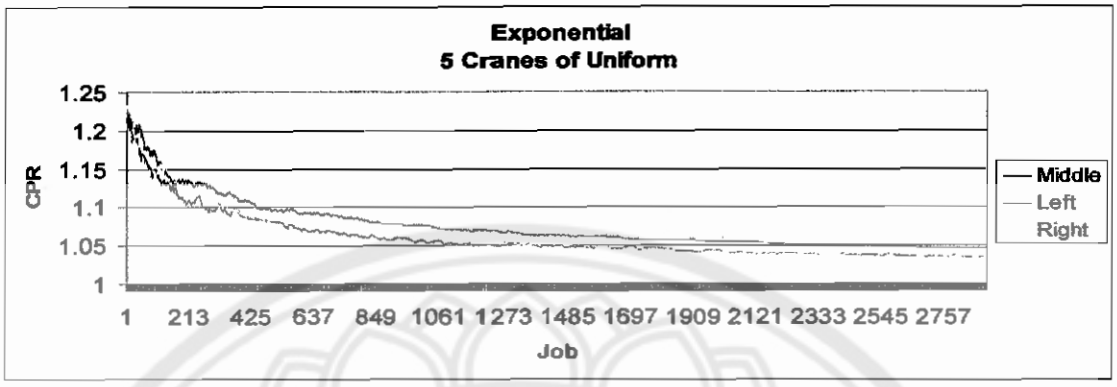
รูปที่ ก-17 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ Exponential



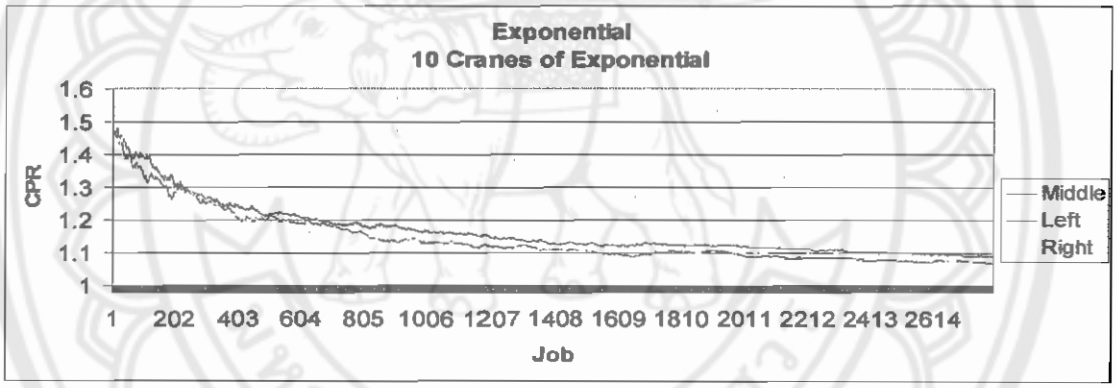
รูปที่ ก-18 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ Exponential



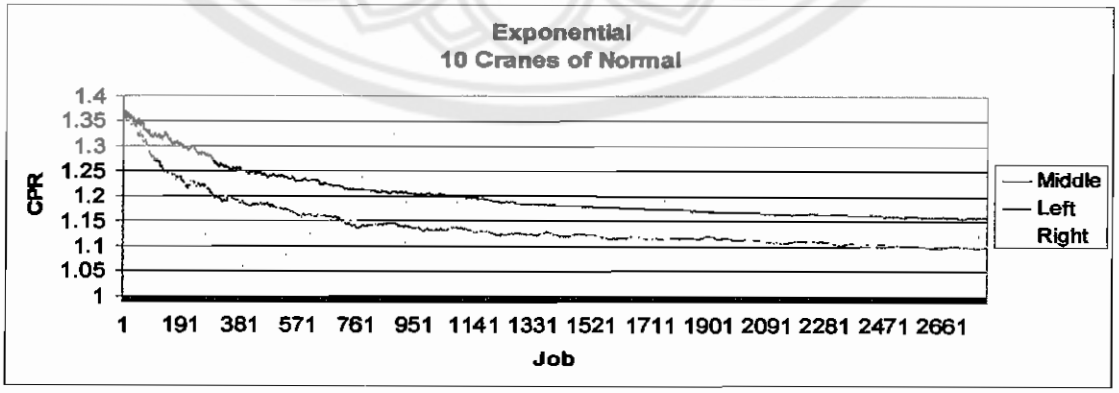
รูปที่ ก-19 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ Exponential



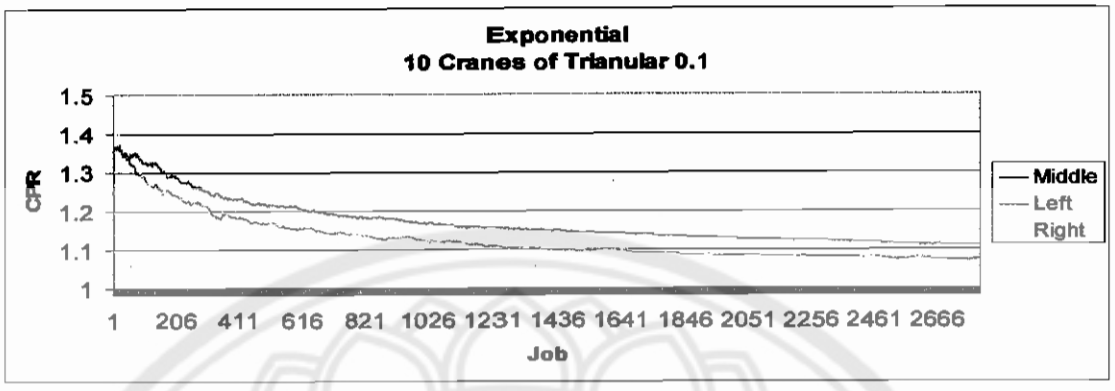
รูปที่ ก-20 แสดงกราฟ การแจกแจงเวลาแบบ Uniform ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ Exponential



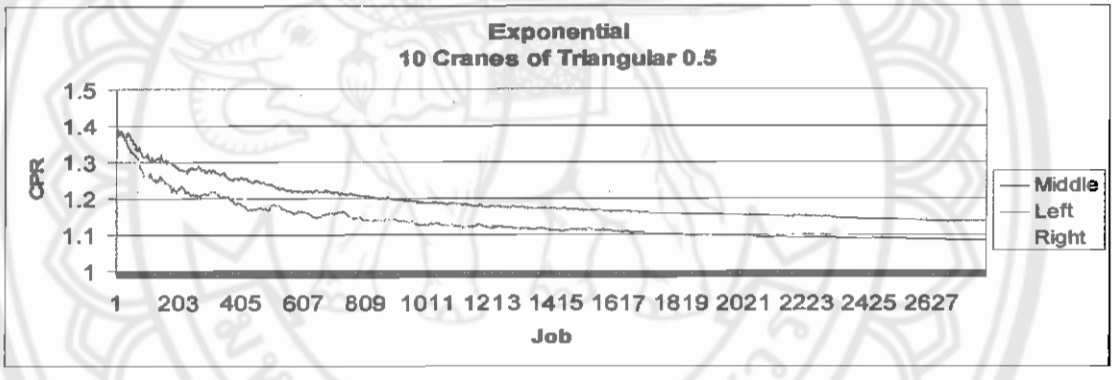
รูปที่ ก-21 แสดงกราฟ การแจกแจงเวลาแบบ Exponential ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Exponential



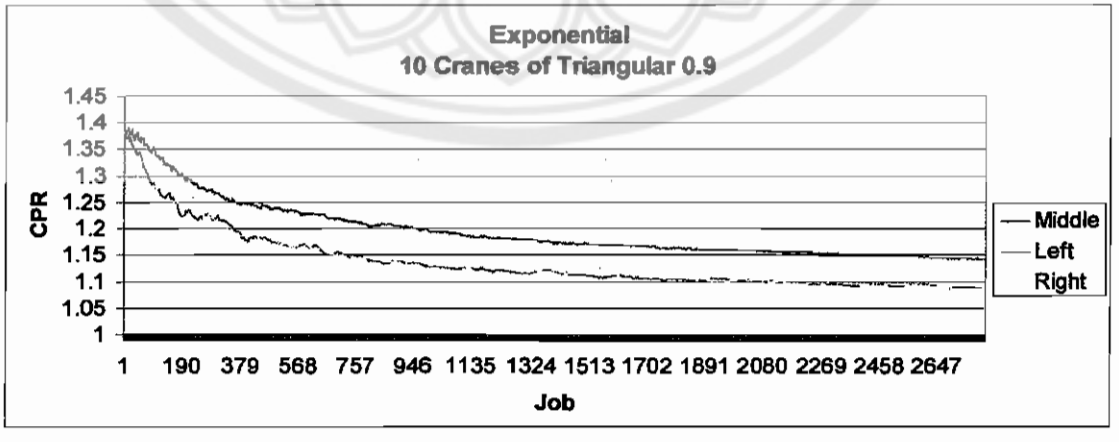
รูปที่ ก-22 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Exponential



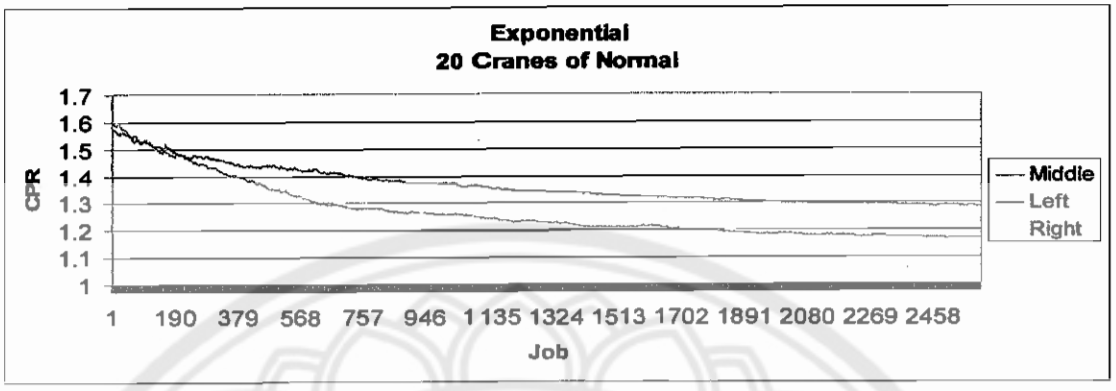
รูปที่ ก-23 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Exponential



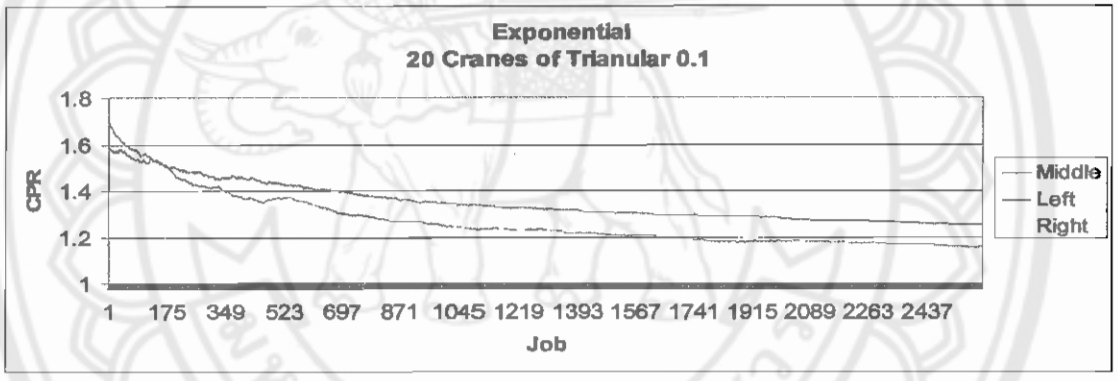
รูปที่ ก-24 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Exponential



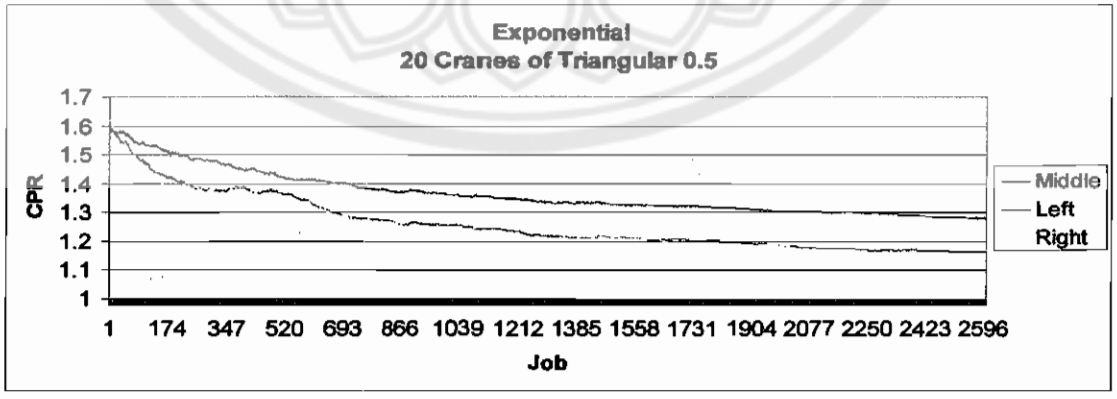
รูปที่ ก-25 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Exponential



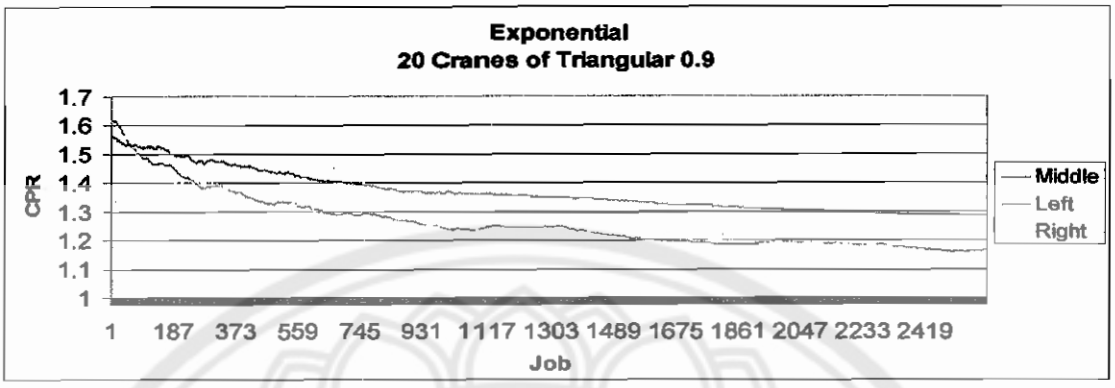
รูปที่ ก-26 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Exponential



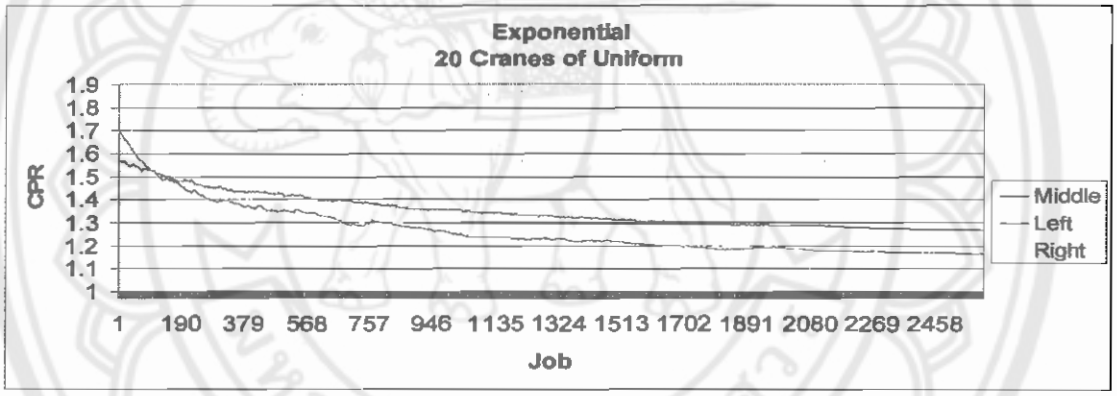
รูปที่ ก-27 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Exponential



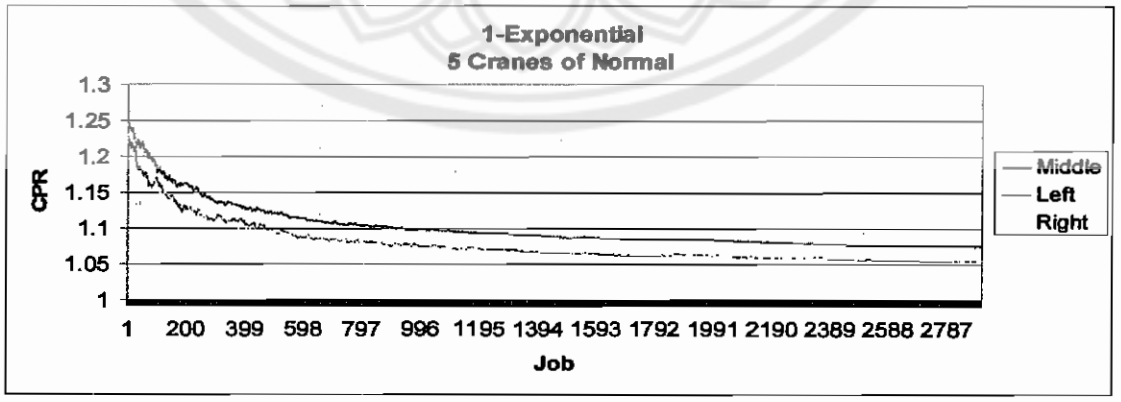
รูปที่ ก-28 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ Exponential



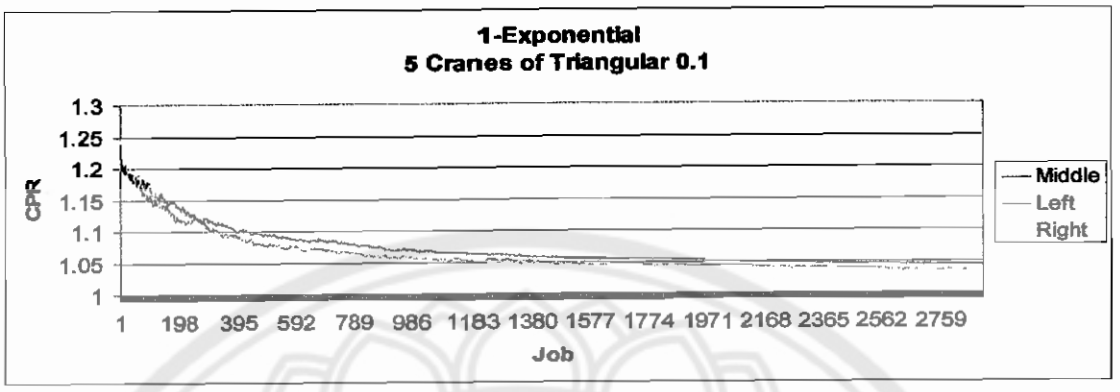
รูปที่ ก-29 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Exponential



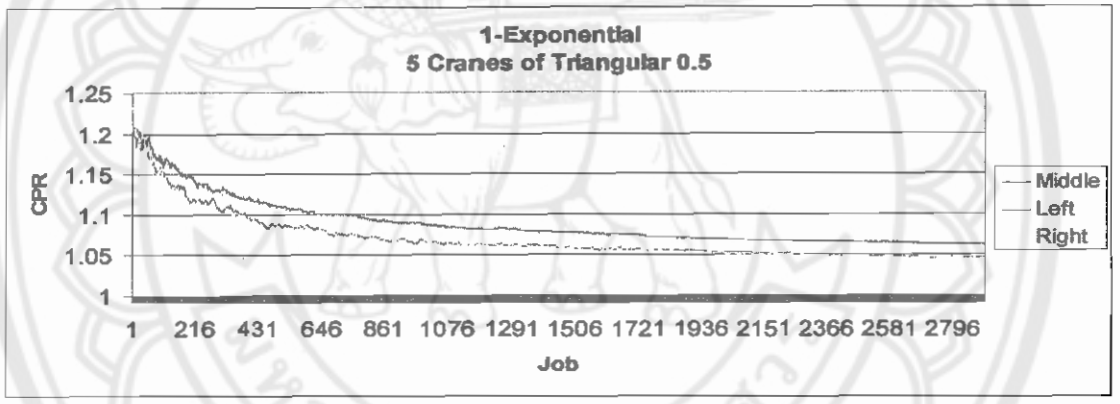
รูปที่ ก-30 แสดงกราฟ การแจกแจงเวลาแบบ Uniform ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ Exponential



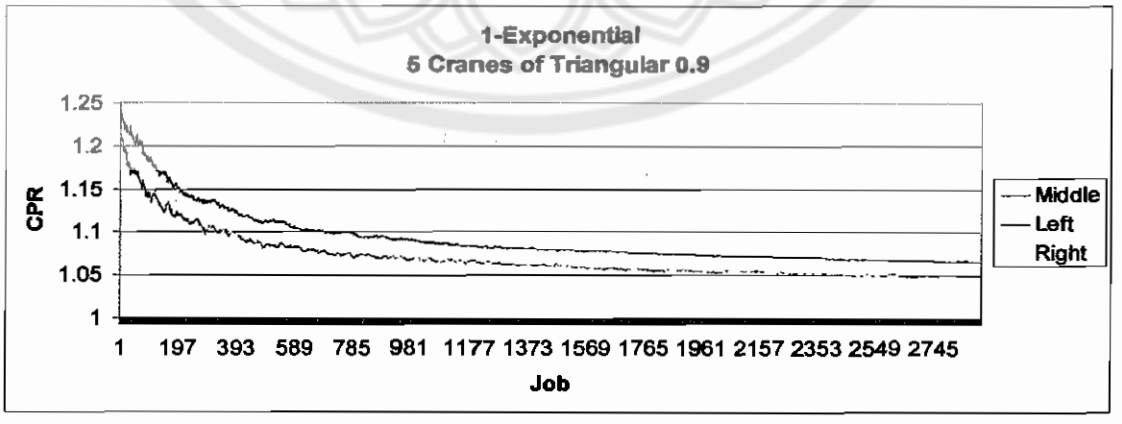
รูปที่ ก-31 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



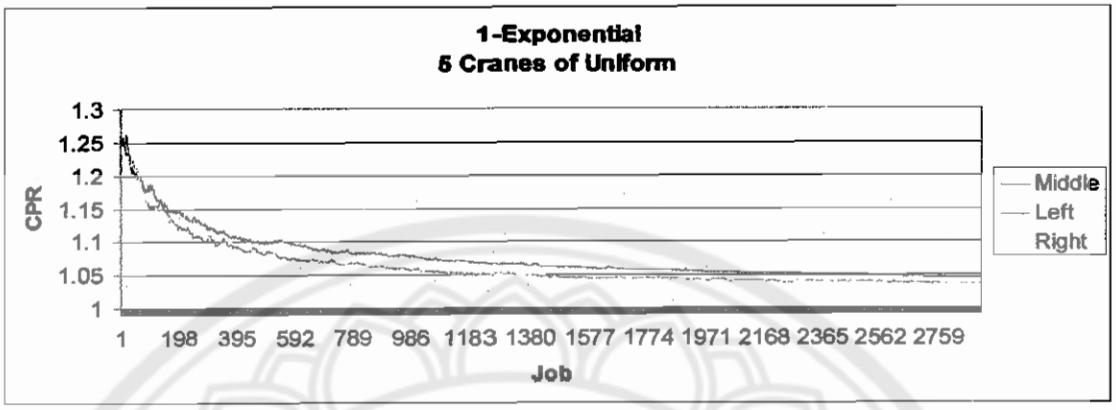
รูปที่ ก-32 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



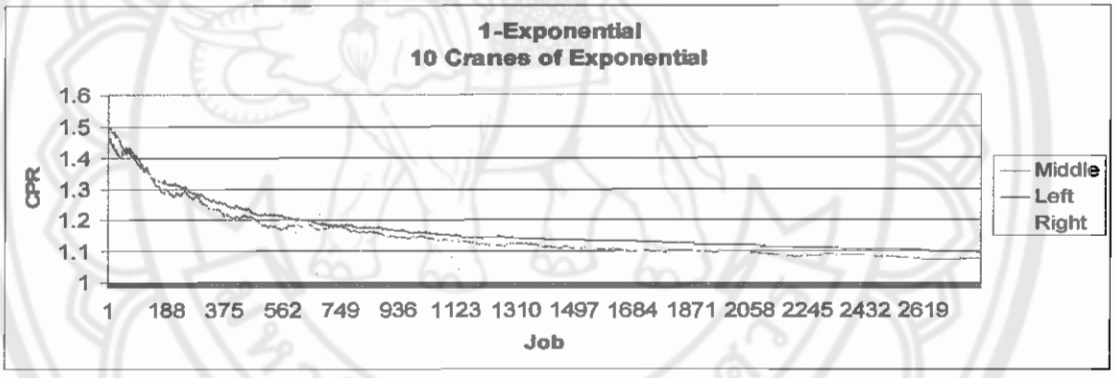
รูปที่ ก-33 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



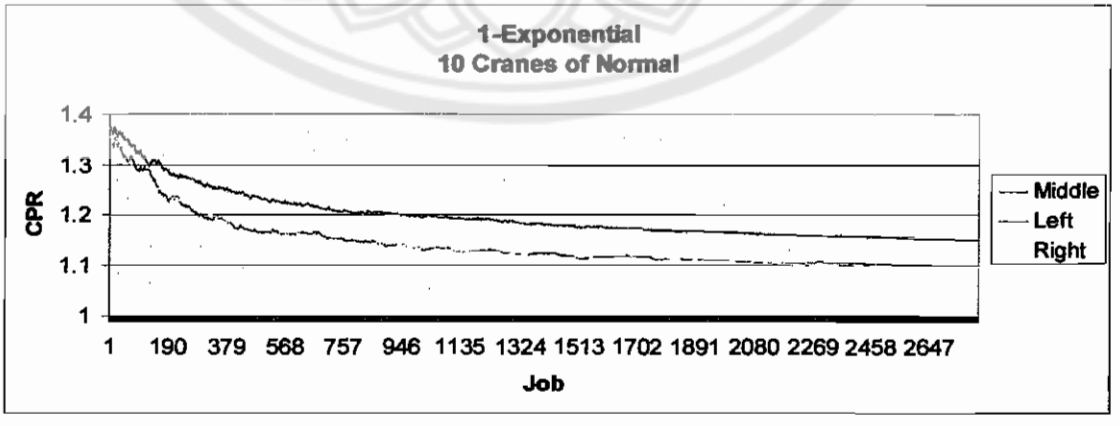
รูปที่ ก-34 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



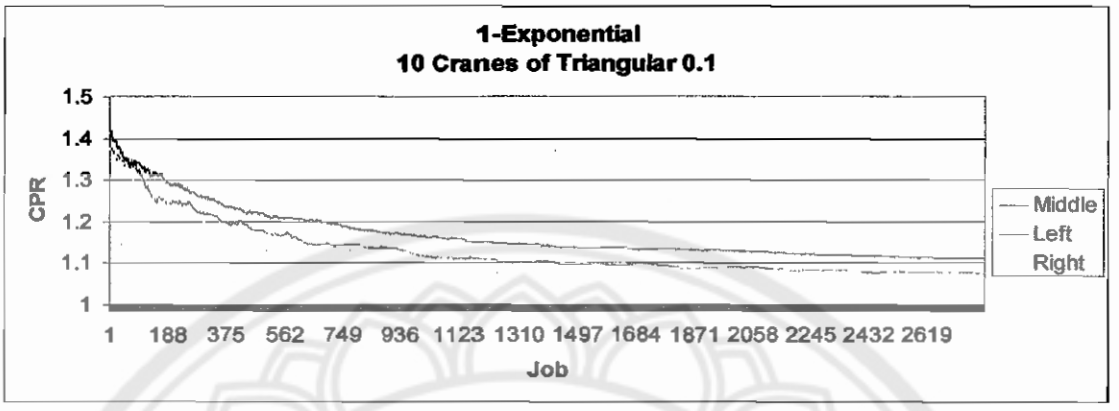
รูปที่ ก-35 แสดงกราฟ การแจกแจงเวลาแบบ Uniform ของจำนวนเครน 5 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



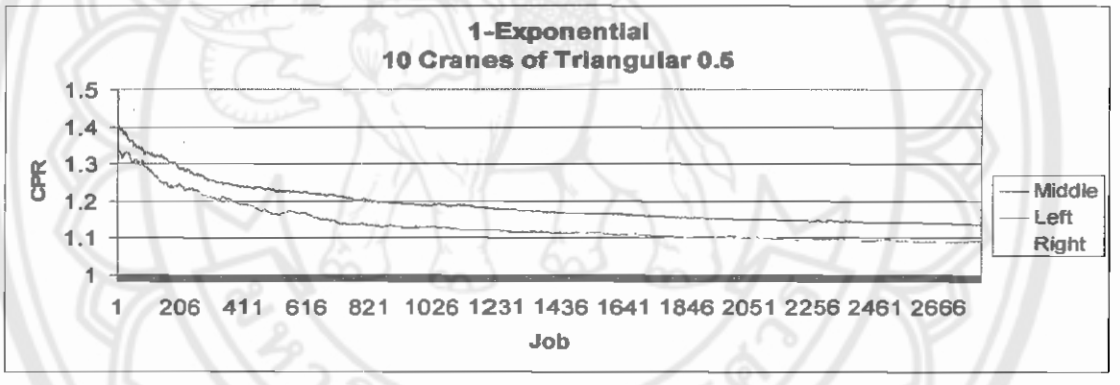
รูปที่ ก-36 แสดงกราฟ การแจกแจงเวลาแบบ Exponential ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



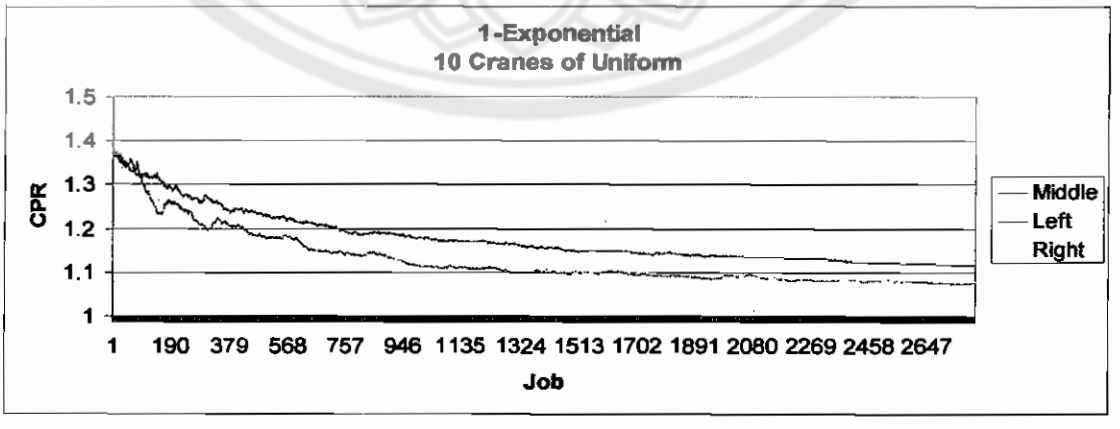
รูปที่ ก-37 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



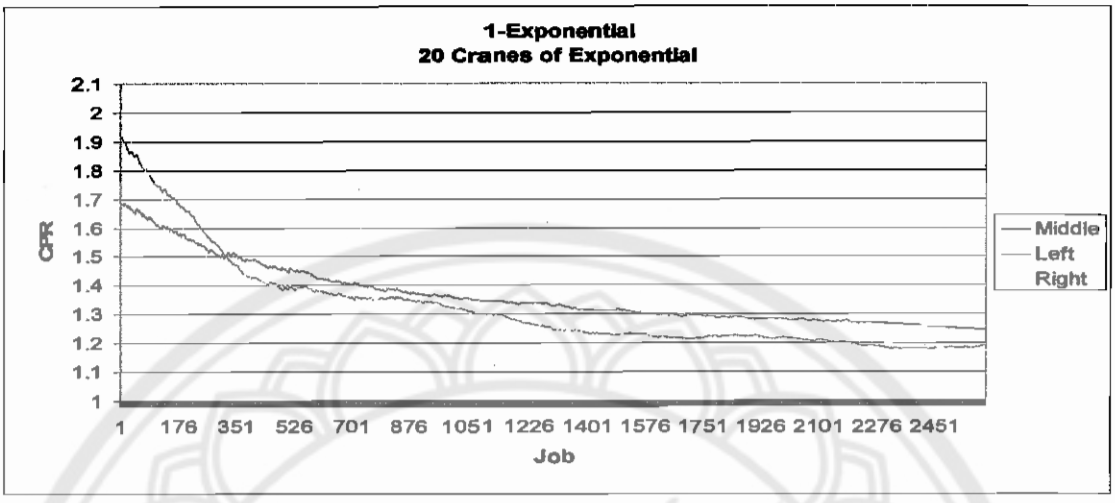
รูปที่ ก-38 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 **ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ** 1-Exponential



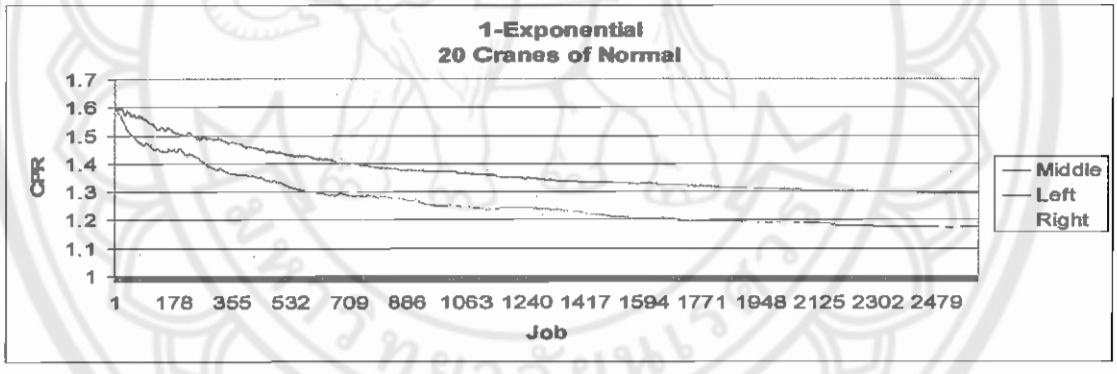
รูปที่ ก-39 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 **ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ** 1-Exponential



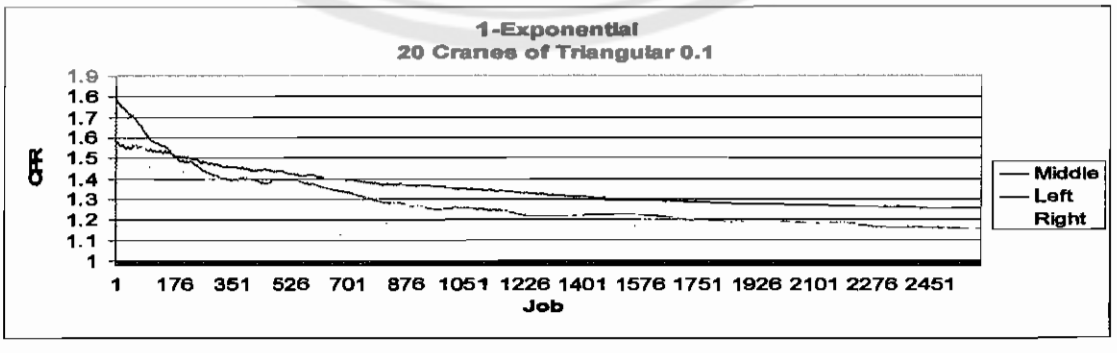
รูปที่ ก-40 แสดงกราฟ การแจกแจงเวลาแบบ Uniform **ของจำนวนเครน 10 เครน และการสุ่มตำแหน่งแบบ** 1-Exponential



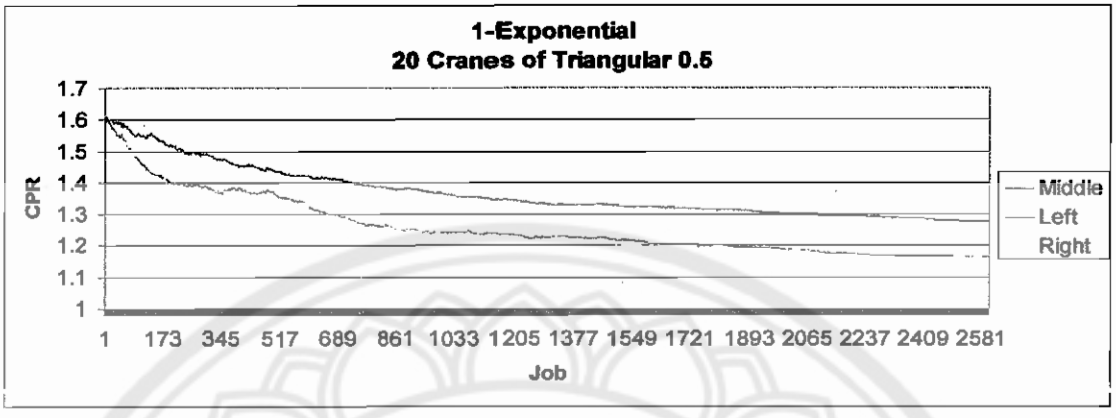
รูปที่ ก-41 แสดงกราฟ การแจกแจงเวลาแบบ Exponential ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



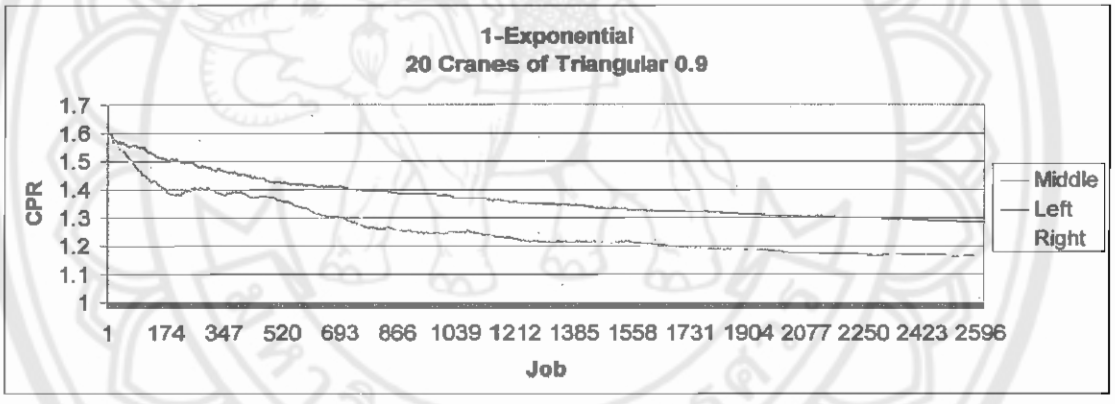
รูปที่ ก-42 แสดงกราฟ การแจกแจงเวลาแบบ Normal ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



รูปที่ ก-43 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.1 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



รูปที่ ก-44 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.5 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ 1-Exponential



รูปที่ ก-45 แสดงกราฟ การแจกแจงเวลาแบบ Triangular 0.9 ของจำนวนเครน 20 เครน และการสุ่มตำแหน่งแบบ 1-Exponential

ตารางผลลัพธ์ที่ได้จากการทดลองโดยวิธีการของ Dudewicz and Dalal
การสุ่มตำแหน่งแบบ Random

ตาราง ก-1 การสุ่มตำแหน่งแบบ Random ของ Exponential 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.067396	0.000020	21	1.062074	4.343682	-3.34368	1.085192
2	1.063967	0.000065	21	1.078093	2.825172	-1.82517	1.038185
3	1.062537	0.000040	21	1.059219	3.34566	-2.34566	1.070318

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-2 การสุ่มตำแหน่งแบบ Random ของ Exponential 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.17162	0.000319	21	1.169828	1.776124	-0.77612	1.173011
2	1.148477	0.00022	21	1.146355	1.954533	-0.95453	1.150503
3	1.140401	0.000158	21	1.129029	2.142424	-1.14242	1.153394

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-3 การสุ่มตำแหน่งแบบ Random ของ Exponential 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.363675	0.00241	21	1.291464	1.177032	-0.17703	1.376459
2	1.313663	0.0014	21	1.326923	1.298206	-0.29821	1.309709
3	1.314057	0.000952	21	1.300484	1.396472	-0.39647	1.319438

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-4 การสุ่มตำแหน่งแบบ Random ของ Normal 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.100574	0.000011	21	1.103867	5.529266	-4.52927	1.085659
2	1.077828	0.000029	21	1.078685	3.766203	-2.7662	1.075458
3	1.079027	0.000012	21	1.081012	5.333983	-4.33398	1.070421

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-5 การสุ่มตำแหน่งแบบ Random ของ Normal 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.199515	0.000054	21	1.189215	3.009332	-2.00933	1.220213
2	1.139429	0.000112	21	1.149606	2.37241	-1.37241	1.125462
3	1.141975	0.00008	21	1.14599	2.637972	-1.63797	1.1354

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-6 การสุ่มตำแหน่งแบบ Random ของ Normal 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.386865	0.00171	21	1.398856	1.251868	-0.25187	1.383845
2	1.255188	0.000435	21	1.256053	1.649168	-0.64917	1.254627
3	1.270423	0.000354	21	1.27345	1.73147	-0.73147	1.268209

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-7 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.1) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.167372	0.000127	21	1.164493	2.283906	-1.28391	1.171068
2	1.126272	0.000136	21	1.119965	2.237927	-1.23793	1.13408
3	1.127895	0.000114	21	1.131709	2.359616	-1.35962	1.12271

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-8 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.1) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.351005	0.000835	21	1.368974	1.43322	-0.43322	1.343221
2	1.272417	0.000797	21	1.271479	1.446741	-0.44674	1.272836
3	1.269458	0.000204	21	1.258983	1.994799	-0.9948	1.279879

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-9 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.5) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.08804	0.000045	21	1.090779	3.207669	-2.20767	1.081991
2	1.069297	0.000033	21	1.06156	3.589119	-2.58912	1.08933
3	1.069178	0.000011	21	1.072568	5.529266	-4.52927	1.053822

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-10 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.5) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.190735	0.000176	21	1.172077	2.07787	-1.07787	1.210846
2	1.13539	0.000093	21	1.132324	2.5137	-1.5137	1.140031
3	1.139605	0.000059	21	1.140513	2.919267	-1.91927	1.137863

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-11 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.5) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.369313	0.000793	21	1.368325	1.448217	-0.44822	1.369756
2	1.25876	0.00028	21	1.272609	1.835207	-0.83521	1.247193
3	1.258609	0.000289	21	1.261243	1.820539	-0.82054	1.256447

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-12 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.9) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.090196	0.000018	21	1.091991	4.527831	-3.52783	1.083863
2	1.070481	0.000031	21	1.066	3.673384	-2.67338	1.082461
3	1.071472	0.000021	21	1.076357	4.261625	-3.26163	1.055542

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-13 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.9) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.191984	0.000085	21	1.215537	2.586828	-1.58683	1.154609
2	1.136243	0.000078	21	1.131764	2.659786	-1.65979	1.143677
3	1.138494	0.000086	21	1.139098	2.577136	-1.57714	1.137542

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-14 การสุ่มตำแหน่งแบบ Random ของ Triangular (0.9) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.361127	0.000788	21	1.363727	1.450077	-0.45008	1.359957
2	1.261248	0.000326	21	1.230673	1.766635	-0.76663	1.284688
3	1.267262	0.000266	21	1.273164	1.859458	-0.85946	1.262189

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-15 การสุ่มตำแหน่งแบบ Random ของ Uniform 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.074907	0.000027	21	1.069423	3.869133	-2.86913	1.090642
2	1.057783	0.000018	21	1.058458	4.527831	-3.52783	1.055403
3	1.059794	0.000027	21	1.061486	3.869133	-2.86913	1.05494

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-16 การสุ่มตำแหน่งแบบ Random ของ Uniform 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.178505	0.000088	21	1.191037	2.558246	-1.55825	1.158977
2	1.128664	0.000057	21	1.124907	2.953874	-1.95387	1.136006
3	1.126822	0.000139	21	1.136416	2.223594	-1.22359	1.115084

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-17 การสุ่มตำแหน่งแบบ Random ของ Uniform 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.35414	0.000649	21	1.377503	1.509576	-0.50958	1.342236
2	1.275437	0.000284	21	1.29064	1.828604	-0.8286	1.26284
3	1.269213	0.000508	21	1.28894	1.592089	-0.59209	1.257532

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

การสุ่มตำแหน่งแบบ Exponential

ตาราง ก-18 การสุ่มตำแหน่งแบบ Exponential ของ Exponential 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.066738	0.000038	21	1.059723	3.40832	-2.40832	1.083633
2	1.062615	0.000024	21	1.055306	4.04698	-3.04698	1.084886
3	1.063475	0.000045	21	1.069697	3.207669	-2.20767	1.049738

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-19 การสุ่มตำแหน่งแบบ Exponential ของ Exponential 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.165526	0.00019	21	1.160823	2.034067	-1.03407	1.170388
2	1.141709	0.000308	21	1.129852	1.79167	-0.79167	1.151095
3	1.146766	0.000227	21	1.136283	1.938252	-0.93825	1.156602

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-20 การสุ่มตำแหน่งแบบ Exponential ของ Exponential 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.358045	0.00121	21	1.386874	1.33382	-0.33382	1.348422
2	1.320255	0.00133	21	1.292351	1.310538	-0.31054	1.328921
3	1.328124	0.000548	21	1.278769	1.565606	-0.56561	1.356039

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-21 การสุ่มตำแหน่งแบบ Exponential ของ Normal 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.098181	0.000019	21	1.095752	4.432125	-3.43213	1.106518
2	1.078673	0.000036	21	1.085781	3.476117	-2.47612	1.061074
3	1.07925	0.000024	21	1.076086	4.04698	-3.04698	1.088891

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-22 การสุ่มตำแหน่งแบบ Exponential ของ Normal 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.362865	0.00065	21	1.412992	1.509084	-0.50908	1.337347
2	1.265191	0.000474	21	1.290098	1.617086	-0.61709	1.249821
3	1.265564	0.000608	21	1.270057	1.530706	-0.53071	1.26318

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-23 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.1) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.073183	0.000018	21	1.069942	4.527831	-3.52783	1.084617
2	1.060894	0.000019	21	1.063661	4.432125	-3.43213	1.051399
3	1.060408	0.000031	21	1.053623	3.673384	-2.67338	1.078548

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-24 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.1) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.168716	0.000138	21	1.164737	2.22832	-1.22832	1.173603
2	1.124559	0.00013	21	1.119803	2.268055	-1.26806	1.13059
3	1.129414	0.000123	21	1.133934	2.305929	-1.30593	1.12351

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-25 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.1) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.346906	0.000477	21	1.342977	1.614777	-0.61478	1.349322
2	1.265685	0.000346	21	1.242115	1.741091	-0.74109	1.283153
3	1.272842	0.000667	21	1.299516	1.500891	-0.50089	1.259481

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-26 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.5) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.089143	0.000019	21	1.083769	4.432125	-3.43213	1.107588
2	1.069483	0.000015	21	1.066519	4.870248	-3.87025	1.080951
3	1.06988	0.000016	21	1.069091	4.745465	-3.74547	1.072835

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-27 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.5) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.189692	0.000104	21	1.207003	2.427198	-1.4272	1.164986
2	1.13258	0.000064	21	1.150112	2.839935	-1.83993	1.100324
3	1.13843	0.000151	21	1.155043	2.170558	-1.17056	1.118984

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-28 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.5) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.363907	0.000756	21	1.351414	1.462386	-0.46239	1.369683
2	1.25824	0.000281	21	1.27742	1.833543	-0.83354	1.242251
3	1.257115	0.000338	21	1.248774	1.751042	-0.75104	1.26338

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-29 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.9) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.091617	0.000023	21	1.091961	4.11385	-3.11385	1.090547
2	1.072798	0.000011	21	1.066338	5.529266	-4.52927	1.102057
3	1.072407	0.000023	21	1.065545	4.11385	-3.11385	1.093775

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-30 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.9) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.194472	0.000117	21	1.196111	2.341039	-1.34104	1.192274
2	1.135606	0.000082	21	1.131883	2.616957	-1.61696	1.141627
3	1.136565	0.000053	21	1.143396	3.028852	-2.02885	1.122704

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-31 การสุ่มตำแหน่งแบบ Exponential ของ Triangular (0.9) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.365455	0.000309	21	1.37285	1.790223	-0.79022	1.359611
2	1.259955	0.000557	21	1.226345	1.560029	-0.56003	1.278778
3	1.260242	0.000304	21	1.277547	1.797527	-0.79753	1.24644

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-32 การสุ่มตำแหน่งแบบ Exponential ของ Uniform 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.19887	0.000029	21	1.195335	3.766203	-2.7662	1.20865
2	1.181291	0.000025	21	1.186924	3.984157	-2.98416	1.164479
3	1.184913	0.000036	21	1.186308	3.476117	-2.47612	1.18146

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-33 การสุ่มตำแหน่งแบบ Exponential ของ Uniform 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.172105	0.000086	21	1.16959	2.577136	-1.57714	1.176071
2	1.126277	0.000114	21	1.12181	2.359616	-1.35962	1.132349
3	1.128601	0.000075	21	1.122148	2.69412	-1.69412	1.139533

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-34 การสุ่มตำแหน่งแบบ Exponential ของ Uniform 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.348539	0.000722	21	1.371655	1.476299	-0.4763	1.337529
2	1.264466	0.00025	21	1.271986	1.889584	-0.88958	1.257776
3	1.281538	0.000731	21	1.275337	1.472527	-0.47253	1.284468

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

การสุ่มตำแหน่งแบบ 1- Exponential

ตาราง ก-35 การสุ่มตำแหน่งแบบ 1-Exponential ของ Exponential 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.068618	0.000037	21	1.066579	3.441533	-2.44153	1.073596
2	1.064493	0.000101	21	1.064948	2.449391	-1.44939	1.063835
3	1.06143	0.000029	21	1.051072	3.766203	-2.7662	1.090084

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-36 การสุ่มตำแหน่งแบบ 1-Exponential ของ Exponential 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.167254	0.000185	21	1.150189	2.049146	-1.04915	1.185159
2	1.144648	0.000213	21	1.148426	1.971599	-0.9716	1.140978
3	1.143013	0.000258	21	1.137047	1.874177	-0.87418	1.148228

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-37 การสุ่มตำแหน่งแบบ 1-Exponential ของ Normal 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.10015	0.000019	21	1.105656	4.432125	-3.43213	1.081254
2	1.077432	0.000028	21	1.079783	3.816292	-2.81629	1.07081
3	1.078532	0.000009	21	1.079377	6.013313	-5.01331	1.074296

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-38 การสุ่มตำแหน่งแบบ 1-Exponential ของ Normal 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.196783	0.000089	21	1.205964	2.549039	-1.54904	1.182563
2	1.140537	0.000137	21	1.1378	2.233098	-1.2331	1.143913
3	1.13956	0.000079	21	1.140091	2.648776	-1.64878	1.138684

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-39 การสุ่มตำแหน่งแบบ 1-Exponential ของ Normal 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.369807	0.000958	21	1.349451	1.394759	-0.39476	1.377843
2	1.258775	0.000413	21	1.247417	1.669172	-0.66917	1.266376
3	1.268134	0.00039	21	1.262349	1.691817	-0.69182	1.272137

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-40 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.1) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.070736	0.000036	21	1.072219	3.476117	-2.47612	1.067064
2	1.060544	0.00002	21	1.056747	4.343682	-3.34368	1.073238
3	1.060206	0.000019	21	1.060625	4.432125	-3.43213	1.058767

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-41 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.1) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.167882	0.000121	21	1.163426	2.317344	-1.31734	1.173752
2	1.128162	0.00016	21	1.108288	2.134723	-1.13472	1.150714
3	1.126034	0.000163	21	1.143058	2.123436	-1.12344	1.106907

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-42 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.1) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.346059	0.00047	21	1.373116	1.620198	-0.6202	1.329277
2	1.277393	0.000538	21	1.277285	1.57196	-0.57196	1.277455
3	1.268201	0.000405	21	1.219015	1.676836	-0.67684	1.301493

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-43 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.5) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.088114	0.000046	21	1.0938	3.1828	-2.1828	1.075702
2	1.069339	0.000033	21	1.071737	3.589119	-2.58912	1.063128
3	1.0692	0.000013	21	1.07953	5.161674	-4.16167	1.026208

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-44 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.5) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.188243	0.00019	21	1.187047	2.034067	-1.03407	1.189479
2	1.133957	0.000043	21	1.136278	3.259979	-2.25998	1.128711
3	1.1368	0.000046	21	1.122639	3.1828	-2.1828	1.167709

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-45 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.5) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.366143	0.000904	21	1.392009	1.410744	-0.41074	1.355518
2	1.257996	0.000227	21	1.289725	1.938252	-0.93825	1.228226
3	1.255101	0.000209	21	1.268675	1.981728	-0.98173	1.241775

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-46 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.9) 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.091594	0.000024	21	1.094584	4.04698	-3.04698	1.082484
2	1.069872	0.000036	21	1.072533	3.476117	-2.47612	1.063285
3	1.072594	0.000035	21	1.064204	3.51217	-2.51217	1.093668

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-47 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.9) 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.193768	0.000105	21	1.204072	2.420011	-1.42001	1.179137
2	1.138189	0.000093	21	1.132648	2.5137	-1.5137	1.146575
3	1.135108	0.000107	21	1.108001	2.405939	-1.40594	1.173219

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-48 การสุ่มตำแหน่งแบบ 1-Exponential ของ Triangular (0.9) 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.36937	0.00048	21	1.340653	1.612489	-0.61249	1.386959
2	1.255446	0.000224	21	1.246979	1.945138	-0.94514	1.263448
3	1.255723	0.000325	21	1.222812	1.767972	-0.76797	1.280998

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-49 การสุ่มตำแหน่งแบบ 1-Exponential ของ Uniform 5 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.07597	0.000024	21	1.072441	4.04698	-3.04698	1.086724
2	1.061871	0.000023	21	1.056924	4.11385	-3.11385	1.077277
3	1.062309	0.000018	21	1.052587	4.527831	-3.52783	1.096605

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-50 การสุ่มตำแหน่งแบบ 1-Exponential ของ Uniform 10 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.176397	0.000092	21	1.181356	2.52232	-1.52232	1.168847
2	1.128496	0.000088	21	1.115861	2.558246	-1.55825	1.148183
3	1.127789	0.000147	21	1.12996	2.187521	-1.18752	1.125211

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตาราง ก-51 การสุ่มตำแหน่งแบบ 1-Exponential ของ Uniform 20 Cranes

i	$\bar{X}_i^{(1)}(20)$	$S_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i-20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$
1	1.351553	0.000938	21	1.381514	1.40053	-0.40053	1.339552
2	1.270274	0.000538	21	1.278721	1.57196	-0.57196	1.265443
3	1.26513	0.000424	21	1.246386	1.658981	-0.65898	1.277481

หมายเหตุ ข้อมูลที่ขีดเส้นใต้แสดงถึงแบบจำลองที่ให้ผลลัพธ์ที่ดีที่สุด

ตารางผลลัพธ์ที่ได้จากผลของการรันโปรแกรมด้วยการเรียงลำดับตำแหน่งของงาน

ตาราง ก-52 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Exponential - 5 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	3.314005111	2.008239061	5
เรียงจากมากไปน้อย	2.650117594	5	2.003944885

ตาราง ก-53 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Exponential - 10 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	5.302083454	2.695831329	10
เรียงจากมากไปน้อย	4.553294559	10	2.690835013

ตาราง ก-54 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Exponential - 20 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	9.098101365	3.604985781	20
เรียงจากมากไปน้อย	8.032956245	20	3.648750322

ตาราง ก-55 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Normal - 10 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	4.543582507	1.895941807	10
เรียงจากมากไปน้อย	3.687905342	10	1.887414077

ตาราง ก-56 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Normal - 20 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	7.427777004	2.193834981	20
เรียงจากมากไปน้อย	6.352714703	20	2.20822526

ตาราง ก-57 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.1 - 5 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	3.183476706	1.796263481	5
เรียงจากมากไปน้อย	2.454996209	5	1.490999242

ตาราง ก-58 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.1 - 10 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	4.893108284	2.258707278	10
เรียงจากมากไปน้อย	4.115082667	10	2.25987701

ตาราง ก-59 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.1 - 20 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	8.20532559	2.786938242	20
เรียงจากมากไปน้อย	7.119592656	20	2.78679807

ตาราง ก-60 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.5 - 5 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	3.07692067	1.632047249	5
เรียงจากมากไปน้อย	2.323018228	5	1.636883083

ตาราง ก-61 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.5 - 10 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	4.637420059	2.016643153	10
เรียงจากมากไปน้อย	3.829723415	10	2.015446963

ตาราง ก-62 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.5 - 20 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	7.633056401	2.308721287	20
เรียงจากมากไปน้อย	6.525870241	20	2.322720879

ตาราง ก-63 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.9 - 5 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	3.052412414	1.597691134	5
เรียงจากมากไปน้อย	2.292133783	5	1.597691134

ตาราง ก-64 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.9 - 10 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	4.584267566	1.900487187	10
เรียงจากมากไปน้อย	3.710817091	10	1.911533298

ตาราง ก-65 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Triangular 0.9 - 20 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	7.456194516	2.151927511	20
เรียงจากมากไปน้อย	6.375191866	20	2.147048215

ตาราง ก-66 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Uniform - 5 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	3.165286175	1.747149827	5
เรียงจากมากไปน้อย	2.437634847	5	1.749994682

ตาราง ก-67 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Uniform - 10 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	4.872841591	2.155639913	10
เรียงจากมากไปน้อย	4.03400593	10	2.146935754

ตาราง ก-68 ค่า CPR เฉลี่ยตัวสุดท้ายของงานของ Uniform - 20 Cranes

ตำแหน่งของงาน	CPR		
	Middle Heuristic	Left Heuristic	Right Heuristic
เรียงจากน้อยไปมาก	8.05757707	2.546652732	20
เรียงจากมากไปน้อย	6.988599309	20	2.546652732

ภาคผนวก ข
โค้ดที่ใช้ในการเขียนโปรแกรม

ภาคผนวก ข

จะกล่าวถึงโค้ดที่ใช้ในการเขียนโปรแกรม

ตารางที่ ข-1 แสดงโค้ดคำสั่งที่ทำการสุ่มการแจกแจงเวลาการทำงานรูปแบบต่าง ๆ

```
Private Sub CommandButton1_Click()
Dim NumJob As Integer
NumJob = Cells(1, 3)
For i = 1 To NumJob
    Cells(6 + i, 1) = i
Next i
Dim NumCrane As Integer
NumCrane = Cells(3, 3)
Dim ProcTime() As Double
ReDim ProcTime(1 To NumJob)
Dim Beta As Double   ' Exponential
Beta = 0.2
  For i = 1 To NumJob
     Cells(6 + i, 2) = -Beta * Log(Rnd)
        If Cells(6 + i, 2) > 1 Then
           Cells(6 + i, 2) = 1
        End If
        ProcTime(i) = Cells(6 + i, 2)
  Next i

Dim U1 As Double    'Normal
Dim U2 As Double
Dim V1 As Double
Dim V2 As Double
Dim W As Double
Dim Y As Double
```

```
Dim X1 As Double
Dim X2 As Double
Dim X3 As Double
Dim X4 As Double
  For i = 1 To NumJob / 2
    Do
      U1 = Rnd
      U2 = Rnd
      V1 = 2 * U1 - 1
      V2 = 2 * U2 - 1
      W = V1 ^ 2 + V2 ^ 2
    Loop While W > 1
    Y = Sqr((-2 * Log(W)) / W)
    X1 = V1 * Y
    X2 = V2 * Y
    X3 = X1 / 6 + 0.5
    X4 = X2 / 6 + 0.5
    If X3 < 0 Or X3 > 1 Then
      X3 = 0.5
    End If
    If X4 < 0 Or X4 > 1 Then
      X4 = 0.5
    End If
    Cells(2 * i - 1 + 6, 2) = X3
    Cells(2 * i + 6, 2) = X4
  Next i
  For i = 1 To NumJob
    ProcTime(i) = Cells(6 + i, 2)
  Next i
```

```
Dim RN  As Double    ' Triangular 0.1
Dim m As Double
   m = 0.1
   For i = 1 To NumJob
      RN = Rnd
      If RN <= m Then
         Cells(6 + i, 2) = Sqr(m * RN)
      Else
         Cells(6 + i, 2) = 1 - Sqr((1 - m) * (1 - RN))
      End If
         ProcTime(i) = Cells(6 + i, 2)
   Next i

m = 0.5       ' Triangular 0.5
For i = 1 To NumJob
      RN = Rnd
      If RN <= m Then
         Cells(6 + i, 2) = Sqr(m * RN)
      Else
         Cells(6 + i, 2) = 1 - Sqr((1 - m) * (1 - RN))
      End If
         ProcTime(i) = Cells(6 + i, 2)
Next i

m = 0.9       ' Triangular 0.9
For i = 1 To NumJob
      RN = Rnd
      If RN <= m Then
         Cells(6 + i, 2) = Sqr(m * RN)
```

```
        Else
            Cells(6 + i, 2) = 1 - Sqr((1 - m) * (1 - RN))
        End If
            ProcTime(i) = Cells(6 + i, 2)
Next i
Dim a As Double          ' Uniform
Dim b As Double
    a = 0
    b = 1
    For i = 1 To NumJob
        Cells(6 + i, 2) = a + (b - a) * Rnd
        ProcTime(i) = Cells(6 + i, 2)
    Next i
End Sub
```

ตาราง ข-2 แสดงโค้ดคำสั่งที่ทำการสุ่มตำแหน่งของงานในรูปแบบต่าง ๆ

```
Private Sub CommandButton1_Click()
Dim Pos() As Double
ReDim Pos(1 To NumJob)
    For i = 1 To NumJob      ' Random
        Cells(6 + i, 3) = Rnd
        Pos(i) = Cells(6 + i, 3)
    Next i
Dim Beta As Double           ' Exponential
    Beta = 0.2
    For i = 1 To NumJob
    Cells(6 + i, 3) = -Beta * Log(Rnd)
        If Cells(6 + i, 3) > 1 Then
            Cells(6 + i, 3) = Cells(6 + i, 3) - 1
```

```
    End If
  Pos(i) = Cells(6 + i, 3)
  Next i

Beta = 0.2                    '1 - Exponential
For i = 1 To NumJob
  Cells(6 + i, 3) = 1 - (-Beta * Log(Rnd))
    If Cells(6 + i, 3) >= 0 And Cells(6 + i, 3) <= 1 Then
      Pos(i) = Cells(6 + i, 3)
    End If
  Next i
End Sub
```

ตาราง ซ-3 แสดงโค้ดคำสั่งใช้ล้างข้อมูลเก่าก่อนทำการประมวณผลการทดลองใหม่

```
Private Sub CommandButton2_Click()
  Sheets("Sheet1").Range("A7").CurrentRegion.Clear
  Sheets("Sheet1").Range("B7").CurrentRegion.Clear
  Sheets("Sheet1").Range("C7").CurrentRegion.Clear
  Sheets("Sheet2").Range("B4").CurrentRegion.Clear
  Sheets("Sheet2").Range("C4").CurrentRegion.Clear
  Sheets("Sheet2").Range("D4").CurrentRegion.Clear
  Sheets("Sheet2").Range("F4").CurrentRegion.Clear
  Sheets("Sheet2").Range("G4").CurrentRegion.Clear
  Sheets("Sheet2").Range("H4").CurrentRegion.Clear
  Sheets("Sheet2").Range("J4").CurrentRegion.Clear
  Sheets("Sheet2").Range("K4").CurrentRegion.Clear
  Sheets("Sheet2").Range("L4").CurrentRegion.Clear
End Sub
```

ตาราง ข-4 แสดงโค้ดคำสั่งให้ทำการจัดลำดับการทำงานของเครน เพื่อหาคำตอบของการทดลอง

```
Dim ST() As Double
Dim FT() As Double
Dim NumJobCrane() As Double
Dim NumInCrane() As Double
Dim JST() As Double
Dim GAP() As Double
Dim STG() As Double
Dim FTG() As Double
Dim crane() As Double
Dim PosCrane() As Double
Dim BestTime() As Double
Dim BJST() As Double
Dim OrderInCrane() As Integer
Dim JSTF() As Double
Dim JFTF() As Double
Dim JFT() As Double
Dim LWST() As Double
ReDim JFT(1 To NumJob, 1 To NumCrane)
ReDim OrderInCrane(1 To NumJob)  ' Order of job i on the best crane Crane(i)
ReDim JSTF(1 To NumJob) 'Best Starting time of job i
ReDim JFTF(1 To NumJob) 'Best Finishing time of job i
ReDim ST(1 To NumJob, 1 To NumCrane) 'Starting time of the ith job in crane j
ReDim FT(1 To NumJob, 1 To NumCrane) 'Finishing time of the ith job in crane j
ReDim NumJobCrane(1 To NumJob, 1 To NumCrane) 'Job Number of the ith job in
crane j
ReDim NumInCrane(1 To NumCrane) ' Number of jobs in crane i
ReDim JST(1 To NumJob, 1 To NumCrane) 'Current job starting time of the ith job in
```

```
crane j
ReDim GAP(1 To NumCrane) 'Number of gaps in crane i
ReDim STG(1 To NumJob, 1 To NumCrane) 'Starting time of the ith GAP in crane j
ReDim FTG(1 To NumJob, 1 To NumCrane) 'Finishing time of the ith GAP in crane j
ReDim crane(1 To NumJob) ' Crane Number assigned to process job i
ReDim PosCrane(1 To NumJob, 1 To NumCrane) ' Position of the ith job in crane j
ReDim BestTime(1 To NumCrane) 'The best time for each crane to start processing the
job
ReDim BJST(1 To NumJob, 1 To NumCrane)
ReDim LWST(1 To NumJob)
'Middle Heuristics
    For i = 1 To NumCrane
      NumInCrane(i) = 0
    Next i
    For i = 1 To NumCrane
      GAP(i) = 1
    Next i
    For i = 1 To NumCrane
      STG(1, i) = 0
      FTG(1, i) = 10000
    Next i
Dim Makespan() As Double
ReDim Makespan(0 To NumJob)
Makespan(0) = 0
Dim LB As Double
Dim CPR As Double
Dim Sum As Double
    Sum = 0
    For i = 1 To NumJob
```

```
For j = 1 To NumCrane
  For k = 1 To GAP(j)
    BJST(i, j) = STG(k, j)
    For l = 1 To NumCrane
      JST(i, j) = BJST(i, j)
      If l <> j Then
        yes = 1
        If NumInCrane(l) > 0 Then
          If JST(i, j) > FT(NumInCrane(l), l) Then
            yes = 1
          End If
          If JST(i, j) + ProcTime(i) > FTG(k, j) Then
            yes = 0
            Exit For 'l
            Exit For 'k
          End If
          For m = 1 To NumInCrane(l)
            If JST(i, j) < FT(m, l) And JST(i, j) + ProcTime(i) > ST(m, l) Then
              If j < l Then
                If Pos(i) > PosCrane(m, l) Then
                  yes = 0
                  If FT(m, l) + ProcTime(i) <= FTG(k, j) Then 'Fixing starts
here
                    JST(i, j) = FT(m, l)
                    If m < NumInCrane(l) Then
                      If ST(m + 1, l) < FTG(k, j) Then
                        If JST(i, j) + ProcTime(i) <= ST(m + 1, l) Then
                          yes = 1
                          Exit For 'm
```

```
                End If
            End If
        ElseIf m = NumInCrane(l) Then
            JST(i, j) = FT(m, l)
            JFT(i, j) = JST(i, j) + ProcTime(i)
            If JST(i, j) + ProcTime(j) <= FTG(k, j) Then
                yes = 1
            Else
                yes = 0
            End If
        End If
    Else
        Exit For 'm
        Exit For 'l
        Exit For 'k
    End If
Else
    yes = 1
End If
ElseIf j > l Then
    If Pos(i) < PosCrane(m, l) Then
    yes = 0
    If FT(m, l) + ProcTime(i) <= FTG(k, j) Then
        JST(i, j) = FT(m, l)
        If m < NumInCrane(l) Then
            If ST(m + 1, l) < FTG(k, j) Then
                If JST(i, j) + ProcTime(i) <= ST(m + 1, l) Then
                    yes = 1
                    Exit For 'm
```

```
                End If
            End If
        ElseIf m = NumInCrane(l) Then
            JST(i, j) = FT(m, l)
            JFT(i, j) = JST(i, j) + ProcTime(i)
            If JST(i, j) + ProcTime(i) <= FTG(k, j) Then
                yes = 1
            Else
                yes = 0
            End If
        End If
    Else
        Exit For 'm
        Exit For 'l
        Exit For 'k
    End If
Else
    yes = 1
End If
End If
ElseIf JST(i, j) + ProcTime(i) < ST(m, l) Then
    Exit For
End If
Next m
ElseIf NumInCrane(l) = 0 Then
    yes = 1
End If
If yes = 1 Then
    If JST(i, j) > BJST(i, j) Then
```

```
                    I = 0
                    BJST(i, j) = JST(i, j)
                End If
            ElseIf yes = 0 Then
                Exit For 'I
            End If
        End If
    Next I
    If yes = 1 Then
        Exit For 'k
    End If
  Next k
 Next j
'Now we have job i start time for every crane
Dim CraneList() As Integer
ReDim CraneList(1 To NumCrane)
Dim Count As Integer
  LWST(i) = 10000
  Count = 0
For j = 1 To NumCrane
  If JST(i, j) < LWST(i) Then
    LWST(i) = JST(i, j)
  End If
Next j
For j = 1 To NumCrane
  If LWST(i) = JST(i, j) Then
    Count = Count + 1
    CraneList(Count) = j
  End If
```

```
        Next j
        Dim MiddleOrder As Integer
        If Count Mod 2 = 0 Then
           MiddleOrder = (Count / 2) + 1
           Else
              MiddleOrder = (Count / 2) + 0.5
        End If

        Dim Middle As Integer
           Middle = CraneList(MiddleOrder)
        JSTF(i) = JST(i, Middle) 'start time of the current job
        JFTF(i) = JST(i, Middle) + ProcTime(i) ' Finish time of the current job
        crane(i) = Middle 'Crane assigned to the current job
        If JFTF(i) < Makespan(i - 1) Then
           Makespan(i) = Makespan(i - 1)
        Else
           Makespan(i) = JFTF(i)
        End If
        Sheets("Sheet2").Cells(3 + i, 2) = Makespan(i)
        Sum = Sum + ProcTime(i)
        LB = Sum / NumCrane
        Sheets("Sheet2").Cells(3 + i, 3) = LB
        CPR = Makespan(i) / LB
        Sheets("Sheet2").Cells(3 + i, 4) = CPR
'Update Job
        Dim Temp1 As Double
        Dim Temp2 As Double
        Dim Temp3 As Double
        Dim Temp4 As Double
```

```
Dim TempCrane1 As Double
Dim TempCrane2 As Double
If NumInCrane(Middle) > 0 Then
   Count = 1
   If ST(NumInCrane(Middle), Middle) > JST(i, Middle) Then
      For a = 1 To NumInCrane(Middle)
         If ST(a, Middle) > JST(i, Middle) Then
            If Count = 1 Then
               Temp1 = ST(a, Middle)
               ST(a, Middle) = JST(i, Middle)
               Temp3 = FT(a, Middle)
               FT(a, Middle) = JST(i, Middle) + ProcTime(i)
               TempCrane1 = PosCrane(a, Middle)
               PosCrane(a, Middle) = Pos(i)
               OrderInCrane(i) = a
            End If
               Count = 0
            If Count = 0 Then
               Temp2 = Temp1
               Temp1 = ST(a + 1, Middle)
               ST(a + 1, Middle) = Temp2
               Temp4 = Temp3
               Temp3 = FT(a + 1, Middle)
               FT(a + 1, Middle) = Temp4
               TempCrane2 = TempCrane1
               TempCrane1 = PosCrane(a + 1, Middle)
               PosCrane(a + 1, Middle) = TempCrane2
               OrderInCrane(i) = a + 1
            End If
```

```
            End If
        Next a
        NumInCrane(Middle) = NumInCrane(Middle) + 1
    ElseIf ST(NumInCrane(Middle), Middle) <= JST(i, Middle) Then 'no need for
equal sign
        NumInCrane(Middle) = NumInCrane(Middle) + 1
        ST(NumInCrane(Middle), Middle) = JST(i, Middle)
        FT(NumInCrane(Middle), Middle) = JST(i, Middle) + ProcTime(i)
        PosCrane(NumInCrane(Middle), Middle) = Pos(i)
        OrderInCrane(i) = NumInCrane(Middle)
    End If
  Else
    ST(1, Middle) = JST(i, Middle)
    FT(1, Middle) = JST(i, Middle) + ProcTime(i)
    PosCrane(1, Middle) = Pos(i)
    OrderInCrane(i) = 1
    NumInCrane(Middle) = NumInCrane(Middle) + 1
  End If
'Update GAP
  GAP(Middle) = 0
  For b = 1 To NumJob
    STG(b, Middle) = 0
    FTG(b, Middle) = 0
  Next b
    If ST(1, Middle) > 0 Then
      STG(1, Middle) = 0
      FTG(1, Middle) = ST(1, Middle)
      GAP(Middle) = GAP(Middle) + 1
    End If
```

```
        If NumInCrane(Middle) > 1 Then
          For c = 1 To NumInCrane(Middle) - 1
            If ST(c + 1, Middle) - FT(c, Middle) > 0 Then
              GAP(Middle) = GAP(Middle) + 1
              STG(GAP(Middle), Middle) = FT(c, Middle)
              FTG(GAP(Middle), Middle) = ST(c + 1, Middle)
            End If
          Next c
        End If
        GAP(Middle) = GAP(Middle) + 1 'fixed to GAP(Middle)
        STG(GAP(Middle), Middle) = FT(NumInCrane(Middle), Middle)
        FTG(GAP(Middle), Middle) = 10000
    Next i
    'Left Heuristics
    For i = 1 To NumCrane
        NumInCrane(i) = 0
    Next i
    For i = 1 To NumCrane
      GAP(i) = 1
    Next i
    For i = 1 To NumCrane
      STG(1, i) = 0
      FTG(1, i) = 10000
    Next i
    Sum = 0
    For i = 1 To NumJob
        For j = 1 To NumCrane
          For k = 1 To GAP(j)
            BJST(i, j) = STG(k, j)
```

```
For l = 1 To NumCrane
  JST(i, j) = BJST(i, j)
 If l <> j Then
    yes = 1
    If NumInCrane(l) > 0 Then
      If JST(i, j) > FT(NumInCrane(l), l) Then
        yes = 1
      End If
      If JST(i, j) + ProcTime(i) > FTG(k, j) Then
        yes = 0
        Exit For 'l
        Exit For 'k
      End If
      For m = 1 To NumInCrane(l)
        If JST(i, j) < FT(m, l) And JST(i, j) + ProcTime(i) > ST(m, l) Then
          If j < l Then
            If Pos(i) > PosCrane(m, l) Then
              yes = 0
              If FT(m, l) + ProcTime(i) <= FTG(k, j) Then 'Fixing starts
here
                JST(i, j) = FT(m, l)
                If m < NumInCrane(l) Then
                  If ST(m + 1, l) < FTG(k, j) Then
                    If JST(i, j) + ProcTime(i) <= ST(m + 1, l) Then
                      yes = 1
                      Exit For 'm
                    End If
                  End If
                ElseIf m = NumInCrane(l) Then
```

```
                    JST(i, j) = FT(m, l)
                    JFT(i, j) = JST(i, j) + ProcTime(i)
                    If JST(i, j) + ProcTime(j) <= FTG(k, j) Then
                      yes = 1
                    Else
                      yes = 0
                    End If
                  End If
                Else
                  Exit For 'm
                  Exit For 'l
                  Exit For 'k
                End If
              Else
                yes = 1
              End If
            ElseIf j > l Then
              If Pos(i) < PosCrane(m, l) Then
                yes = 0
                If FT(m, l) + ProcTime(i) <= FTG(k, j) Then
                  JST(i, j) = FT(m, l)
                  If m < NumInCrane(l) Then
                    If ST(m + 1, l) < FTG(k, j) Then
                      If JST(i, j) + ProcTime(i) <= ST(m + 1, l) Then
                        yes = 1
                        Exit For 'm
                      End If
                    End If
                  ElseIf m = NumInCrane(l) Then
```

```
                    JST(i, j) = FT(m, l)
                    JFT(i, j) = JST(i, j) + ProcTime(i)
                    If JST(i, j) + ProcTime(i) <= FTG(k, j) Then
                      yes = 1
                    Else
                      yes = 0
                    End If
                  End If
                Else
                  Exit For 'm
                  Exit For 'l
                  Exit For 'k
                End If
              Else
                yes = 1
              End If
            End If
          ElseIf JST(i, j) + ProcTime(i) < ST(m, l) Then
            Exit For
          End If
        Next m
      ElseIf NumInCrane(l) = 0 Then
        yes = 1
      End If
      If yes = 1 Then
        If JST(i, j) > BJST(i, j) Then
          l = 0
          BJST(i, j) = JST(i, j)
        End If
```

```
                ElseIf yes = 0 Then
                    Exit For 'l
                End If
            End If
        Next l
        If yes = 1 Then
            Exit For 'k
        End If
    Next k
Next j

LWST(i) = 10000
Count = 0
For j = 1 To NumCrane
    If JST(i, j) < LWST(i) Then
        LWST(i) = JST(i, j)
    End If
Next j
For j = 1 To NumCrane
    If LWST(i) = JST(i, j) Then
        Count = Count + 1
        CraneList(Count) = j
    End If
Next j
Dim LeftOrder As Integer
If Count > 0 Then
    LeftOrder = (Count / Count)
End If
Dim Left As Integer
```

```
        Left = CraneList(LeftOrder)
    JSTF(i) = JST(i, Left) 'start time of the current job
    JFTF(i) = JST(i, Left) + ProcTime(i) ' Finish time of the current job
    crane(i) = Left 'Crane assigned to the current job
    If JFTF(i) < Makespan(i - 1) Then
        Makespan(i) = Makespan(i - 1)
    Else
        Makespan(i) = JFTF(i)
    End If
    Sheets("Sheet2").Cells(3 + i, 6) = Makespan(i)
    Sum = Sum + ProcTime(i)
    LB = Sum / NumCrane
    Sheets("Sheet2").Cells(3 + i, 7) = LB
    CPR = Makespan(i) / LB
    Sheets("Sheet2").Cells(3 + i, 8) = CPR
'Update Job
    If NumInCrane(Left) > 0 Then
        Count = 1
        If ST(NumInCrane(Left), Left) > JST(i, Left) Then
            For a = 1 To NumInCrane(Left)
                If ST(a, Left) > JST(i, Left) Then
                    If Count = 1 Then
                        Temp1 = ST(a, Left)
                        ST(a, Left) = JST(i, Left)
                        Temp3 = FT(a, Left)
                        FT(a, Left) = JST(i, Left) + ProcTime(i)
                        TempCrane1 = PosCrane(a, Left)
                        PosCrane(a, Left) = Pos(i)
                        OrderInCrane(i) = a
```

```
            End If
              Count = 0
            If Count = 0 Then
              Temp2 = Temp1
              Temp1 = ST(a + 1, Left)
              ST(a + 1, Left) = Temp2
              Temp4 = Temp3
              Temp3 = FT(a + 1, Left)
              FT(a + 1, Left) = Temp4
              TempCrane2 = TempCrane1
              TempCrane1 = PosCrane(a + 1, Left)
              PosCrane(a + 1, Left) = TempCrane2
              OrderInCrane(i) = a + 1
            End If
          End If
        Next a
        NumInCrane(Left) = NumInCrane(Left) + 1
      ElseIf ST(NumInCrane(Left), Left) <= JST(i, Left) Then 'no need for equal sign
        NumInCrane(Left) = NumInCrane(Left) + 1
        ST(NumInCrane(Left), Left) = JST(i, Left)
        FT(NumInCrane(Left), Left) = JST(i, Left) + ProcTime(i)
        PosCrane(NumInCrane(Left), Left) = Pos(i)
        OrderInCrane(i) = NumInCrane(Left)
      End If
    Else
      ST(1, Left) = JST(i, Left)
      FT(1, Left) = JST(i, Left) + ProcTime(i)
      PosCrane(1, Left) = Pos(i)
      OrderInCrane(i) = 1
```

```
        NumInCrane(Left) = NumInCrane(Left) + 1
    End If
'Update GAP
    GAP(Left) = 0
    For b = 1 To NumJob
        STG(b, Left) = 0
        FTG(b, Left) = 0
    Next b
        If ST(1, Left) > 0 Then
            STG(1, Left) = 0
            FTG(1, Left) = ST(1, Left)
            GAP(Left) = GAP(Left) + 1
        End If
        If NumInCrane(Left) > 1 Then
            For c = 1 To NumInCrane(Left) - 1
                If ST(c + 1, Left) - FT(c, Left) > 0 Then
                    GAP(Left) = GAP(Left) + 1
                    STG(GAP(Left), Left) = FT(c, Left)
                    FTG(GAP(Left), Left) = ST(c + 1, Left)
                End If
            Next c
        End If
        GAP(Left) = GAP(Left) + 1
        STG(GAP(Left), Left) = FT(NumInCrane(Left), Left)
        FTG(GAP(Left), Left) = 10000
Next i
'Right Heuristics
For i = 1 To NumCrane
        NumInCrane(i) = 0
```

```
    Next i
    For i = 1 To NumCrane
      GAP(i) = 1
    Next i
    For i = 1 To NumCrane
      STG(1, i) = 0
      FTG(1, i) = 10000
    Next i
Sum = 0
For i = 1 To NumJob
    For j = 1 To NumCrane
      For k = 1 To GAP(j)
        BJST(i, j) = STG(k, j)
        For l = 1 To NumCrane
          JST(i, j) = BJST(i, j)
        If l <> j Then
          yes = 1
          If NumInCrane(l) > 0 Then
            If JST(i, j) > FT(NumInCrane(l), l) Then
              yes = 1
            End If
          If JST(i, j) + ProcTime(i) > FTG(k, j) Then
            yes = 0
            Exit For 'l
            Exit For 'k
          End If
          For m = 1 To NumInCrane(l)
            If JST(i, j) < FT(m, l) And JST(i, j) + ProcTime(i) > ST(m, l) Then
              If j < l Then
```

```
                If Pos(i) > PosCrane(m, l) Then
                  yes = 0
                  If FT(m, l) + ProcTime(i) <= FTG(k, j) Then 'Fixing starts
here
                    JST(i, j) = FT(m, l)
                    If m < NumInCrane(l) Then
                      If ST(m + 1, l) < FTG(k, j) Then
                        If JST(i, j) + ProcTime(i) <= ST(m + 1, l) Then
                          yes = 1
                          Exit For 'm
                        End If
                      End If
                    ElseIf m = NumInCrane(l) Then
                      JST(i, j) = FT(m, l)
                      JFT(i, j) = JST(i, j) + ProcTime(i)
                      If JST(i, j) + ProcTime(j) <= FTG(k, j) Then
                        yes = 1
                      Else
                        yes = 0
                      End If
                    End If
                  Else
                    Exit For 'm
                    Exit For 'l
                    Exit For 'k
                  End If
                Else
                  yes = 1
                End If
```

```
ElseIf j > l Then
  If Pos(i) < PosCrane(m, l) Then
    yes = 0
    If FT(m, l) + ProcTime(i) <= FTG(k, j) Then
      JST(i, j) = FT(m, l)
      If m < NumInCrane(l) Then
        If ST(m + 1, l) < FTG(k, j) Then
          If JST(i, j) + ProcTime(i) <= ST(m + 1, l) Then
            yes = 1
            Exit For 'm
          End If
        End If
      ElseIf m = NumInCrane(l) Then
        JST(i, j) = FT(m, l)
        JFT(i, j) = JST(i, j) + ProcTime(i)
        If JST(i, j) + ProcTime(i) <= FTG(k, j) Then
          yes = 1
        Else
          yes = 0
        End If
      End If
    Else
      Exit For 'm
      Exit For 'l
      Exit For 'k
    End If
  Else
    yes = 1
  End If
```

```
                End If
                ElseIf JST(i, j) + ProcTime(i) < ST(m, l) Then
                  Exit For
                End If
              Next m
            ElseIf NumInCrane(i) = 0 Then
              yes = 1
            End If
            If yes = 1 Then
              If JST(i, j) > BJST(i, j) Then
                l = 0
                BJST(i, j) = JST(i, j)
              End If
            ElseIf yes = 0 Then
              Exit For 'l
            End If
          End If
        Next l
        If yes = 1 Then
          Exit For 'k
        End If
      Next k
    Next j

LWST(i) = 10000
  Count = 0
For j = 1 To NumCrane
  If JST(i, j) < LWST(i) Then
    LWST(i) = JST(i, j)
```

```
        End If
    Next j
    For j = 1 To NumCrane
        If LWST(i) = JST(i, j) Then
            Count = Count + 1
            CraneList(Count) = j
        End If
    Next j
    Dim RightOrder As Integer
    If Count > 0 Then
        RightOrder = Count
    End If
    Dim Right As Integer
        Right = CraneList(RightOrder)
    JSTF(i) = JST(i, Right) 'start time of the current job
    JFTF(i) = JST(i, Right) + ProcTime(i) ' Finish time of the current job
    crane(i) = Right 'Crane assigned to the current job
    If JFTF(i) < Makespan(i - 1) Then
        Makespan(i) = Makespan(i - 1)
    Else
        Makespan(i) = JFTF(i)
    End If
    Sheets("Sheet2").Cells(3 + i, 10) = Makespan(i)
    Sum = Sum + ProcTime(i)
    LB = Sum / NumCrane
    Sheets("Sheet2").Cells(3 + i, 11) = LB
    CPR = Makespan(i) / LB
    Sheets("Sheet2").Cells(3 + i, 12) = CPR
'Update Job
```

```
If NumInCrane(Right) > 0 Then
  Count = 1
  If ST(NumInCrane(Right), Right) > JST(i, Right) Then
    For a = 1 To NumInCrane(Right)
      If ST(a, Right) > JST(i, Right) Then
        If Count = 1 Then
          Temp1 = ST(a, Right)
          ST(a, Right) = JST(i, Right)
          Temp3 = FT(a, Right)
          FT(a, Right) = JST(i, Right) + ProcTime(i)
          TempCrane1 = PosCrane(a, Right)
          PosCrane(a, Right) = Pos(i)
          OrderInCrane(i) = a
        End If
          Count = 0
        If Count = 0 Then
          Temp2 = Temp1
          Temp1 = ST(a + 1, Right)
          ST(a + 1, Right) = Temp2
          Temp4 = Temp3
          Temp3 = FT(a + 1, Right)
          FT(a + 1, Right) = Temp4
          TempCrane2 = TempCrane1
          TempCrane1 = PosCrane(a + 1, Right)
          PosCrane(a + 1, Right) = TempCrane2
          OrderInCrane(i) = a + 1
        End If
      End If
    Next a
```

```
            NumInCrane(Right) = NumInCrane(Right) + 1
        ElseIf ST(NumInCrane(Right), Right) <= JST(i, Right) Then 'no need for equal
sign
            NumInCrane(Right) = NumInCrane(Right) + 1
            ST(NumInCrane(Right), Right) = JST(i, Right)
            FT(NumInCrane(Right), Right) = JST(i, Right) + ProcTime(i)
            PosCrane(NumInCrane(Right), Right) = Pos(i)
            OrderInCrane(i) = NumInCrane(Right)
        End If
    Else
        ST(1, Right) = JST(i, Right)
        FT(1, Right) = JST(i, Right) + ProcTime(i)
        PosCrane(1, Right) = Pos(i)
        OrderInCrane(i) = 1
        NumInCrane(Right) = NumInCrane(Right) + 1
    End If
'Update GAP
    GAP(Right) = 0
    For b = 1 To NumJob
        STG(b, Right) = 0
        FTG(b, Right) = 0
    Next b
        If ST(1, Right) > 0 Then
            STG(1, Right) = 0
            FTG(1, Right) = ST(1, Right)
            GAP(Right) = GAP(Right) + 1
        End If
        If NumInCrane(Right) > 1 Then
            For c = 1 To NumInCrane(Right) - 1
```

```
                If ST(c + 1, Right) - FT(c, Right) > 0 Then
                    GAP(Right) = GAP(Right) + 1
                    STG(GAP(Right), Right) = FT(c, Right)
                    FTG(GAP(Right), Right) = ST(c + 1, Right)
                End If
            Next c
        End If
        GAP(Right) = GAP(Right) + 1
        STG(GAP(Right), Right) = FT(NumInCrane(Right), Right)
        FTG(GAP(Right), Right) = 10000
Next i

End Sub
```

ภาคผนวก ค
ตารางที่อ้างอิงค่าในการคำนวณ

ตาราง ค-1 ตารางของการหาค่า h_1

P*	n_0	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
0.90	20	1.896	2.342	2.583	2.747	2.870	2.969	3.051	3.121	3.182
0.90	40	1.852	2.283	2.514	2.669	2.785	2.878	2.954	3.019	3.076
0.95	20	2.453	2.872	3.101	3.258	3.377	3.472	3.551	3.619	3.679
0.95	40	2.386	2.786	3.003	3.150	3.260	3.349	3.422	3.484	3.539