**Basic Factory Dynamics (1)**

**<Experiment result & analysis>**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * **Find rb, T0, W0 of the current line.**   Bottleneck Rate (rb): Rate (parts/unit time or jobs/unit time) of the process center having the highest long-term utilization.  Raw Process Time (T0): Sum of the long-term average process times of each station in the line  Critical WIP : W0 = rb T0  이므로, 문제에서 주어진 상황을 활용해 각 값을 구하면 아래와 같다.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Workstation 1 | Workstation 2 | Workstation 3 | Workstation 4 | | # of machine  (resource capacity) | 5 | 12 | 1 | 4 | | Processing time | 15 min/job | 30 min/job | 3 min/job | 10 min/job | | Station Rate  (job / min) |  |  |  |  |   Bottleneck Rate (rb) = min(0.3333, 0.4, 0.3333, 0.4) = 0.3333  Raw Process Time (T0) = 15 + 30 + 3 + 10 = 58 min  Critical WIP (W0) =   * **Verify the Little’s Law by performing the simulation and recording the CT and WIP while changing the production rate (TH) of the current line.**   Production rate(TH)를 변화시키며, CT와 WIP을 관찰하기 위하여 Arena 상에서 아래와 같이 모듈을 배치하고, Create 모듈의 Time between arrival 변화시켰다. TH를 바꾸기 위해 Number Out을 바꾸기 위해 Time between arrival을 바꾼 것이다.    한 번에 비교를 수행하기 위해, “Entry rate”라는 variable을 정의하고 Time between arrival로 설정한 뒤, Process Analyzer을 활용했다.    이 시뮬레이션 결과로 Little’s Law가 성립하는지 확인해보았다.  Little’s Law : The fundamental relation between WIP, CT, and TH over the long-term is WIP = THCT  (TH = Number out Processing Time)    processing time 1000분이 Time between arrival인 entry rate으로 나누어 떨어지는 경우, Little’s Law와 시뮬레이션 결과가 동일하고, 나누어 떨어지지 않는 경우에도 이로 인한 미세한 차이가 있을 뿐이다.  만약 time between arrival이 3분보다 작은, 가령 2분으로 주어지면, bottleneck station인 station 1과 3 중 먼저 있는 station 1에서 병목 현상이 일어나며, 시간이 지날수록 WIP이 무수히 많게 쌓인다. 이로 인해 Arena 실행 시 에러가 발생한다.  station 1의 station rate인 1/3는, 3분에 하나의 작업을 완료할 수 있다는 것인데, time between arrival이 이 값보다 짧게 주어지면 대기열이 계속해서 증가하는 것이다.   * **When adding a new machine, calculate the maximum production rate (THmax) and think of a method to verify it and confirm it using the simulation.**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | TH | Add to Station 1 | Add to Station 2 | Add to Station 3 | Add to Station 4 | | Calculation | 0.3333 | 0.3333 | 0.3333 | 0.3333 | | Simulation | 0.3333 | 0.3333 | 0.3333 | 0.3333 |   Calculation - Add to Station 1   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Workstation 1 | Workstation 2 | Workstation 3 | Workstation 4 | | # of machine | 6 (5 / 1) | 12 | 1 | 4 | | Processing time (min/job) | 15 / 3 | 30 | 3 | 10 | | Station Rate  (job / min) |  |  |  |  |   기존 workstation 1 의 station rate는 이다. 그런데, 기존의 MC보다 성능이 좋은 New MC가 추가됨으로써, station rate가 어떤 양수 만큼 증가할 것이다. 하지만, 여전히 THmax는 bottleneck station(workstation 3)의 station rate인 0.3333이다.  Workstation 2, 3, 4의 경우도 동일한 방식이다.  Calculation - Add to Station 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Workstation 1 | Workstation 2 | Workstation 3 | Workstation 4 | | # of machine | 5 | 13(12 / 1) | 1 | 4 | | Processing time (min/job) | 15 | 30 / 3 | 3 | 10 | | Station Rate  (job / min) |  | + |  |  |   Calculation - Add to Station 3   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Workstation 1 | Workstation 2 | Workstation 3 | Workstation 4 | | # of machine | 5 | 12 | 2(1 / 1) | 4 | | Processing time (min/job) | 15 | 30 | 3 / 3 | 10 | | Station Rate  (job / min) |  |  | +0.3333 |  |   Calculation - Add to Station 4   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Workstation 1 | Workstation 2 | Workstation 3 | Workstation 4 | | # of machine | 5 | 12 | 1 | 5(4 / 1) | | Processing time (min/job) | 15 | 30 | 3 | 10 / 3 | | Station Rate  (job / min) |  |  |  | + |   위 Calculation 결과를 simulation을 통해 검증했다. Little’s Law에 의해, TH = WIP/CT 으로 계산했다. Calculation은 bottleneck station의 Utilization이 1이 되도록 계산되었으므로, Simulation 역시 time between arrival을 3분으로 지정하여 그렇게 되도록 했다.  Simulation - Add to Station 1   |  |  |  | | --- | --- | --- | | WIP | 17.333 |  | | Cycle Time | 52 | | THmax | 0.3333 |   Simulation - Add to Station 2   |  |  |  | | --- | --- | --- | | WIP | 10.333 |  | | Cycle Time | 31 | | THmax | 0.3333 |   Simulation - Add to Station 3   |  |  |  | | --- | --- | --- | | WIP | 19.989 |  | | Cycle Time | 59.9672 | | THmax | 0.3333 |   Simulation - Add to Station 4   |  |  |  | | --- | --- | --- | | WIP | 18.098 |  | | Cycle Time | 54.2892 | | THmax | 0.3333 |   Calculation 값이나, Simulation 결과가 각각 모두 동일하게 나온다. 이는 현재 Line에서, Bottlenect Station이 하나가 아니라 두 개 존재하기 때문이다. 따라서 MC를 새로이 추가하더라도, 전체 공정의 THmax를 개선시키지 못하는 것이다.  참고로, 기존 문제 환경대로 time between arrival을 5분으로 설정하면 simulation 결과가아래와 같이 모두 THmax가 0.2로 동일하게 나온다.   |  |  |  |  | | --- | --- | --- | --- | | Simulation - Add to Station 1 | WIP | 9.2 |  | | Cycle Time | 46 | | THmax | 0.2 | | Simulation - Add to Station 2 | WIP | 6.2 |  | | Cycle Time | 31 | | THmax | 0.2 | | Simulation - Add to Station 3 | WIP | 11.6 |  | | Cycle Time | 58 | | THmax | 0.2 | | Simulation - Add to Station 4 | WIP | 10.2 |  | | Cycle Time | 51 | | THmax | 0.2 | |

**< Discussion & conclusion>**

|  |
| --- |
| **(1) If TH is close to rb, using Little’s Law, identify the relationship between a) WIP and Cycle time of the production line, b) final product inventory and production time, and c) vehicles at the tollgate and their average queue time.**  TH가 rb에 가깝다는 의미는, 생산 라인 내 Bottleneck station의 utilization이 1에 가깝다는 의미이자, “Best Case Performance”를 보인다고 해석된다.  **a) WIP과 Cycle Time의 관계**  Little’s Law에 의해, WIP = THCT이므로 WIP이 증가하면 CT가 비례하게 증가한다. 감소 역시 마찬가지다. 예시로, 실습 문제에서 New MC가 Station 1에 추가됐을 때, TH는 0.3333으로 유지되고 WIP은 19.3333에서 17.3333으로, CT는 58에서 52로 감소했다. 그 비율은 TH의 값인 것을 확인할 수 있다.  추가로, best case에서 TH가 rb에 가까워지면 WIP은 에 가까워지며, Cycle Time은 에 가까워진다.    참고로, 왼쪽 그림과 같이 PWC의 경우에도 TH가 rb에 가까워질 수 있는데, 이는 WIP이 아주 큰 값을 가질 때이다. 이 때, WIP과 Cycle Time은 비례 관계이고, 이에 Cycle Time 역시 큰 값을 가질 것이다. Worst Case에서는 TH가 rb에 가까워질 수 없다. 이 때에는 작은 WIP의 변화가 큰 CT의 증가로 이어진다.  **b) final product inventory와 production time의 관계**  Best Case에서, TH가 rb에 가까워지면 final product inventory는 최적에 가깝게 빠르게 쌓인다. Production time은 위에서 보인 바와 같이 에 가까워진다. PWC에서는 final product inventory는 최적에 가깝게 빠르게 쌓이지만, WIP이 큰 값을 가지고, 이에 따라 Production time 역시 큰 값을 가진다. Worst Case에서는 TH가 에 도달할 수 없으며, final product inventory에 도달하기 위해서 많은 시간이 든다. 마찬가지로 WIP의 증가에 따라 Production time이 민감하게 증가한다.  추가로, production time을 “총 생산라인이 가동되는 시간”으로 해석하면, 아래와 같다.  final product inventory = 으로, TH가 일정하다고 가정하면 final product inventory가 production time에 비례하여 증가한다. 예시로, 실습 예제에서 time between arrival을 3분으로 설정하여 TH = 0.3333인 경우, New MC가 없는 기본 상황에서 1000분의 작동 결과, Number out은 334이다. 임을 확인할 수 있다.  2000분으로 시간을 늘리면 Number out이 664로, 667이다. 비례 관계를 확인할 수 있다.  **c) vehicles at the tollgate와 their average queue time**  Tollgate의 상황을 공정 과정으로 이해하면, 아래와 같다.  WIP = Vehicles at the tollgate  Cycle Time = Average Queue Time (Best case에서, 대기가 발생하지 않는다.)  TH가 rb에 가까워질 때, Best case에서 Vehicles at tollgate는 W0에 가까워지고, Average Queue Time은 T0에 가까워진다.  추가로, Little’s Law에 의해 아래와 같은 관계를 생각할 수 있다.  Vehicles at the tollgate = Average Queue Time TH(# of vehicles passing tollgate per unit time)  Average Queue Time이 길수록 Vehicles at the tollgate가 많은 것이다.  **(2) Production line A has a high WIP and a long cycle time. Production line B has a low WIP and a short cycle time. Is it possible for production line A and production line B to have the same output (TH)? If you were to choose one production line, which one would you choose and why? (Regardless of the experiment result).**  가능하다. Little’s Law에 따르면 WIP = THCT인데, Production line A와 B의 WIP : CT 비율이 같다면, 두 Production line의 TH는 같을 수 있다.  그런데, 둘의 TH가 같다고 하더라도 Production line B를 선택하는 것이 합리적이다. 아래 그림을 통해 이를 설득할 수 있다.    높은 이윤을 얻기 위해, 두 가지 측면에서 고려할 수 있다. 좌측의 Low Cost를 목적으로 삼는 “Factory Plant”와, 우측의 High Sales를 목적으로 삼는 “Marketing & Sales Division”이 있다.  각 측면의 계층적 목표를 살펴보면, 먼저 생산측은 낮은 단가를 위해 높은 생산량과 가동률, 그리고 적은 재고량을 원한다. 이를 위해 적은 변동성과 “짧은 cycle time” 전략을 취한다. 우측 마케팅 및 판매측은 많은 판매량을 위해 높은 품질과 고객 서비스 수준을 원하고, 이를 위해 낮은 가동률, 높은 재고량과 변동성, 그리고 “짧은 cycle time” 전략을 취한다. 두 측면은 변동성, 가동률, 재고 수준에서 서로 다른 전략을 원하지만, “짧은 cycle time”은 둘 모두가 원하는 바이다.  즉, 높은 이윤을 위해서는 동일한 TH를 가진다면 low WIP, short cycle time을 가지는 Production line B를 택하는 것이 합리적이다.  **(3) Discuss the area where Little’s Law can be applied besides the production system.**  Little’s Law는 생산 시스템뿐 아니라 대기열이 있는 다른 시스템에도 적용 가능하다.  - 소매 업장  슈퍼, 의류 매장 등의 업장의 대기열에 있는 고객의 수와 대기 시간은 서비스 수준과 직접적으로 연관되어 있다. 예를 들어, 만약 슈퍼마켓의 대기열의 평균 길이가 10명이고, 평균 대기 시간이 5분이라면, 해당 창구의 직원은 단위 시간당 2명의 고객을 서비스하는 것이다. 콜센터, 은행과 같은 곳에서도 마찬가지로 적용된다. 이를 통해 직원의 업무 능력을 파악할 수도 있고, 대기열을 관리하는 의사결정에 활용할 수 있다. 병목 현상이 일어나는 지점을 파악하여 빠른 회전율을 위하여 자원을 투자한다면 쇼핑을 더 오래하도록 할 지, 창구를 개선할 지 등이 있겠다. 음식점에서도 동일하게 적용된다.  - 의료  병원에서 환자의 흐름을 최적화하는 데 사용된다. 특히, 건강검진과 같이 여러 station을 거쳐야 하는 경우 효과적이다. 환자가 치료를 기다리는 평균 시간, 대기열의 평균 환자 수 등을 분석하여 병목 현상을 식별 및 해결할 수 있다. 예를 들어, 대기열이 긴 bottleneck 부분에 해당하는 검진 센터는 추가 의사 또는 간호사를 고용하거나 예약 시스템을 통해 대기 시간을 줄일 수 있다. 공항에서도 이와 비슷한 원리로 적용되고 있다.  - 웹사이트 서버 관리  웹사이트가 느려지거나 충돌하지 않고 처리할 수 있는 이상적인 동시 사용자 수를 계산하는 데 사용된다. 웹사이트 뿐 아니라 온라인 증권 거래소와 같은 경우, 서버를 운영할 때 사용자의 평균 주문 수와 평균 처리 시간을 분석하여 증권 거래소는 개선이 필요한 영역을 식별하는 것은 필수적이다.  즉, Little’s Law는 생산 시스템 뿐 아니라, 다양한 시스템에 적용되어 개선이 필요한 영역 식별, 수용 능력 향상, 효율적 운영, 고객 서비스 수준 향상 등에 활용되고 있다. |