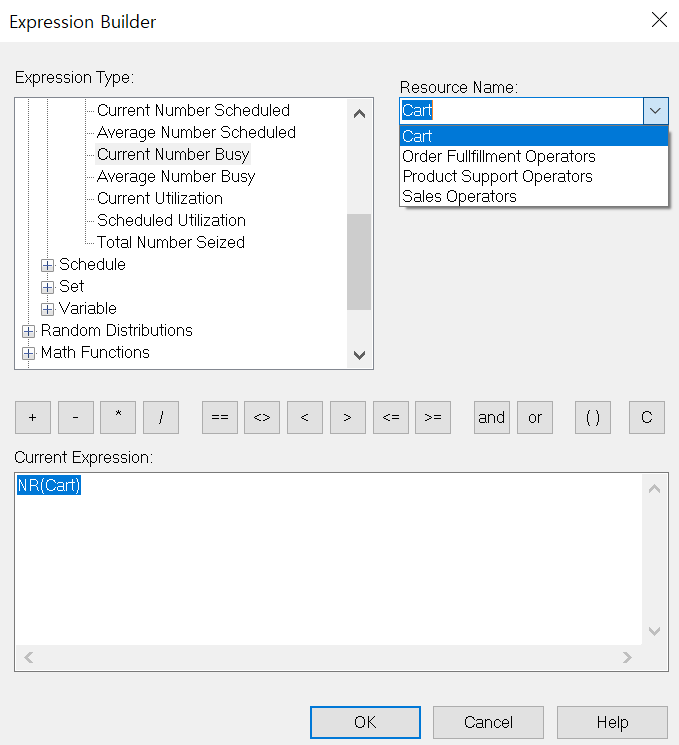
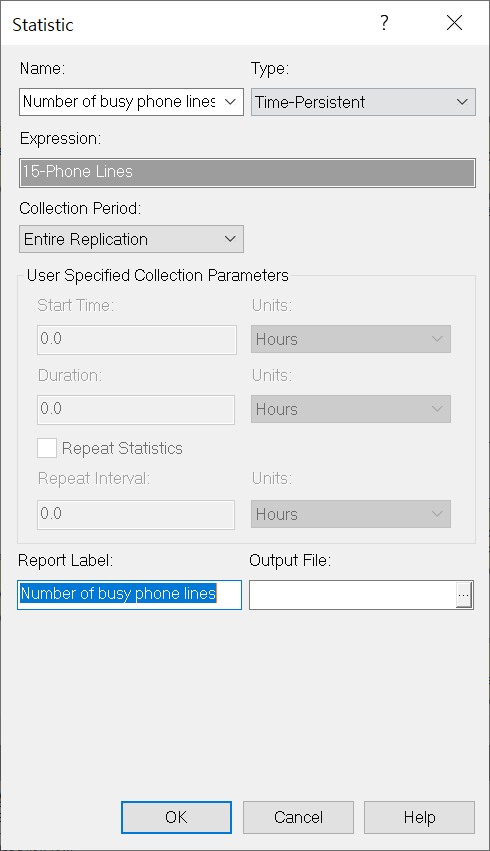
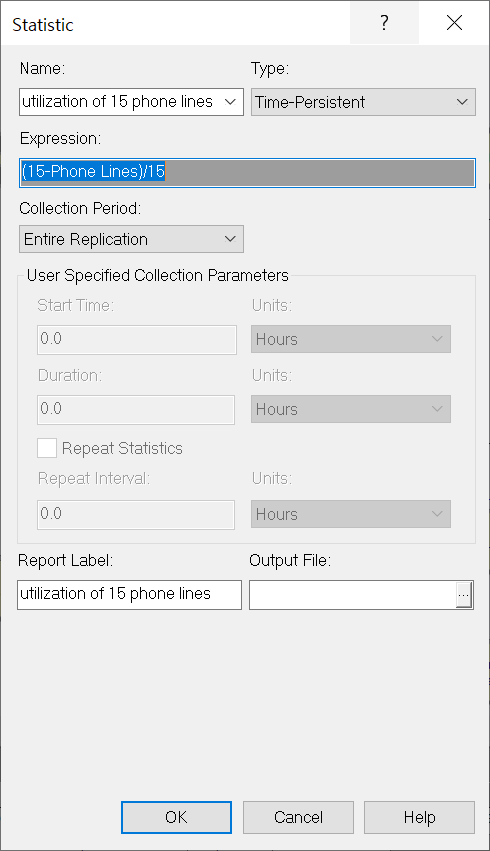
**<1>**



위에서 볼 수 있듯이, Current Number Busy, Current Utilization 등의 Expression은 Resource에서만 표현이 가능하다. 하지만, 주어진 baseline에서는 phone lines를 Resource가 아닌 Variable로 나타내었다. 따라서 아래와 같이 Expression을 구성하여 구하고자 하는 Statistic을 표현했다.



10번의 복제를 기준으로 한 각 95% 신뢰 구간은 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

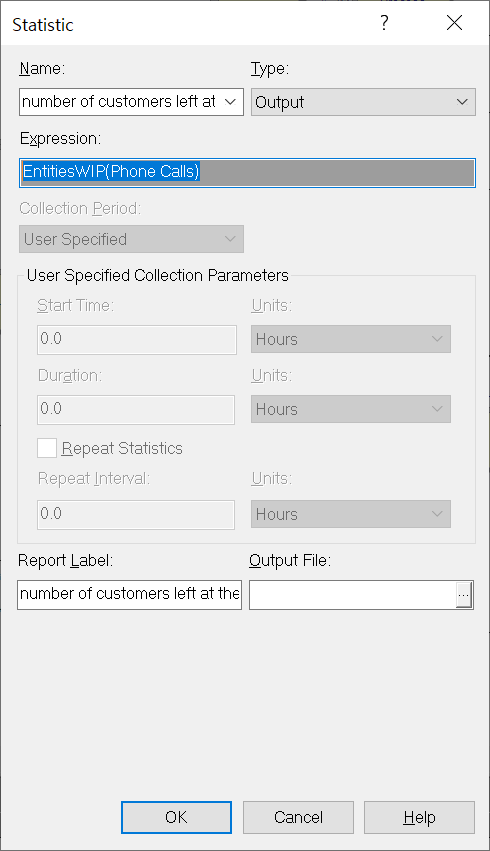
Number of busy phone lines : 8.0055 0.91

**Utilization of 15 phone lines : 0.5337 0.06**

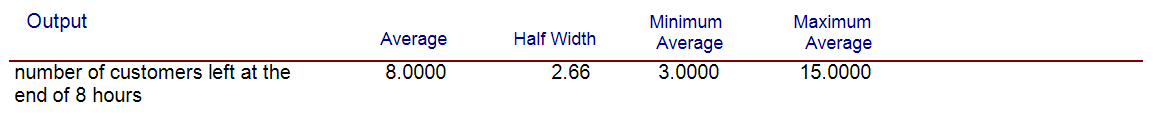
**<2>**

영업 시간인 8시간이 종료되었을 시점에, 남아있는 고객의 수를 얻기 위해서는 문제에 나와 있듯이 Phone Calls의 WIP을 생각하면 된다.

EntitiesWIP(Phone Calls)으로 Phone Calls의 WIP을 구했다.



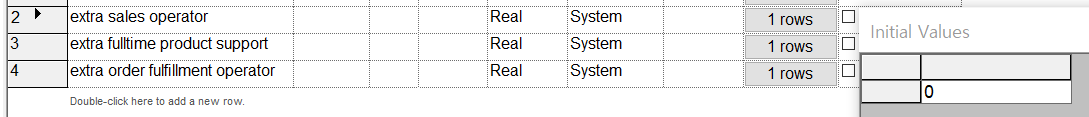
Type을 output으로 설정하여 8시간이 지난 시점에서만 이 statistic을 기록하였고, 10번의 복제를 기준으로 한 각 95% 신뢰 구간은 다음과 같다



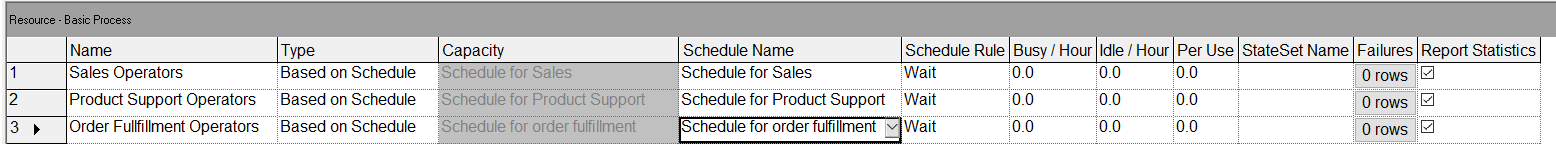
**Number of customers left at the end of 8 hours : 8.0 2.66**

**<3>**

문제의 상황을 구현하기 위하여, extra operator variable을 생성(initial value는 0)했다.



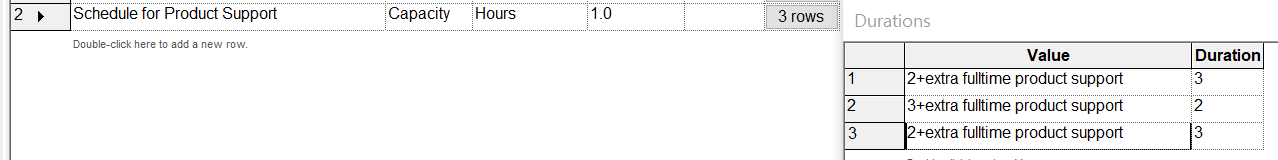
이후에 이를 각 resource의 capacity에 더하는 방식으로 구현했다. 이때, capacity에는 변수를 넣을 수 없으므로 각 schedule을 생성하여 기존의 capacity에 추가 operator을 고려하였다.



텍스트이(가) 표시된 사진

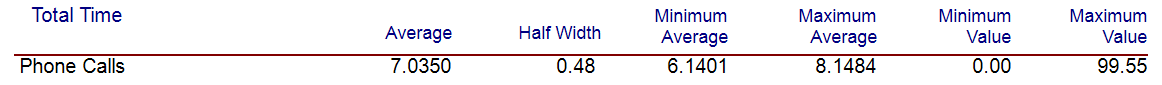
자동 생성된 설명

Product Support Operators의 capacity는 이미 따르고 있는 schedule이 있으므로, schedule에서 extra operator variable을 더해주었다.



테이블이(가) 표시된 사진

자동 생성된 설명

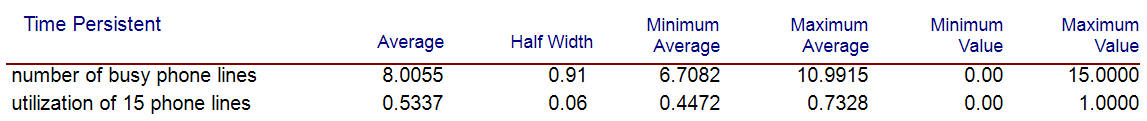


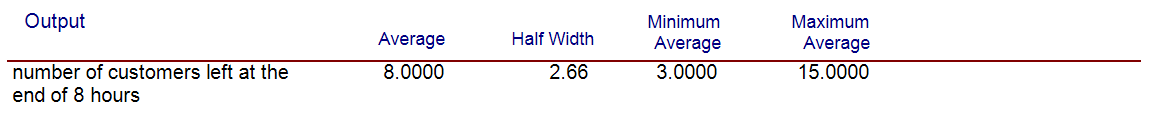
**기존의 overall flowtime의 95% 신뢰구간 : 7.035 0.48**

추가로, ‘performance measures’에서 제시한 또다른 성능 지표인 utilization of operators and phone lines / the number of customers in the call center at the end of the day은 아래와 같다.

테이블이(가) 표시된 사진

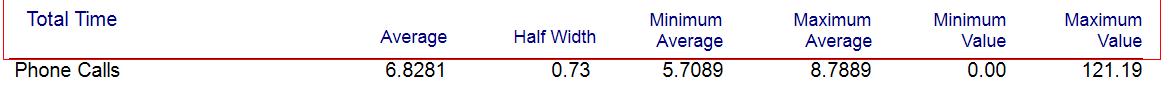
자동 생성된 설명





이제, 각 variable을 차례로 0에서 1로 수정해주며 각 상황에 대한 결과를 구했다.

1. **add an extra sales operator**



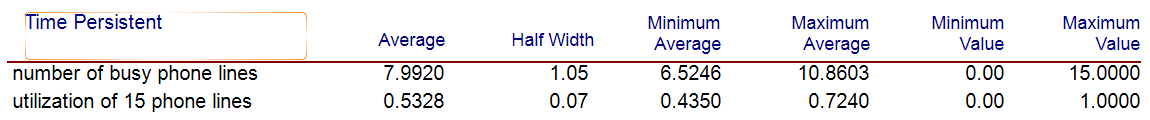
overall flowtime의 95% 신뢰구간 : **6.8281 0.73**

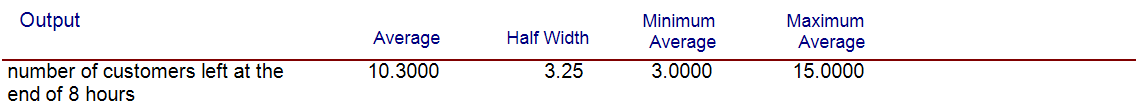
평균은 다소 감소했으나, 신뢰구간 내에서 차이를 보이는 만큼 통계적으로 유의미한 차이가 있다고 볼 수는 없다.

추가로, 다른 성능지표들은 아래와 같다.

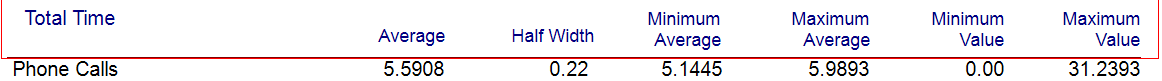
테이블이(가) 표시된 사진

자동 생성된 설명





1. **add an extra full-time product support operator**



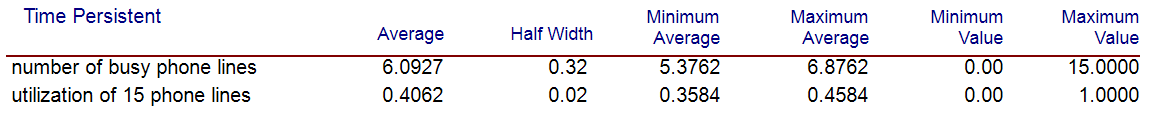
overall flowtime의 95% 신뢰구간 : **5.5908 0.22**

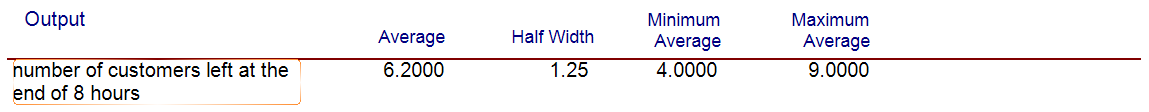
overall flowtime이 통계적으로 유의미하게 감소한 것을 확인할 수 있다.

추가로, 다른 성능지표들은 아래와 같다.

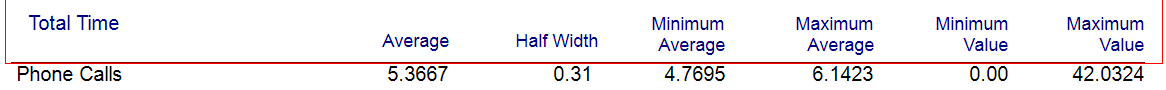
테이블이(가) 표시된 사진

자동 생성된 설명





1. **add an extra order fulfillment operator**



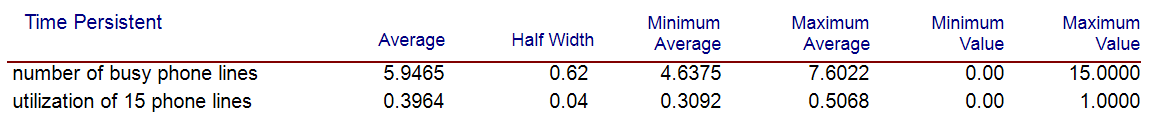
overall flowtime의 95% 신뢰구간 : **5.3667 0.31**

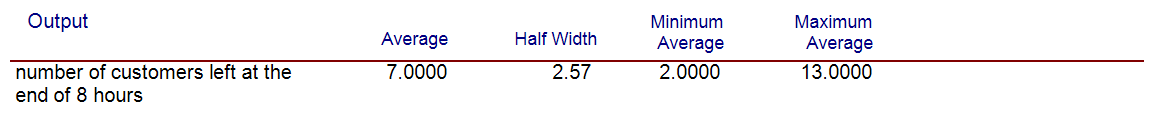
overall flowtime이 통계적으로 유의미하게 감소한 것을 확인할 수 있다.

추가로, 다른 성능지표들은 아래와 같다.

테이블이(가) 표시된 사진

자동 생성된 설명





**<4>**

테이블이(가) 표시된 사진

자동 생성된 설명

위와 같이 data file을 저장하도록 설정하여 <2>에서의 base 경우, <3>에서의 3가지 경우에 대해 각각 파일을 저장했다.

이에 <3>에서 시행한 각 변화 중 flowtime을 가장 크게 감소시킨 변화를 찾기 위해, 아래와 같이 compare means에 데이터를 넣어주었다.

Overall flowtime-base vs. overall flowtime-1

Overall flowtime-base vs. overall flowtime-2

Overall flowtime-base vs. overall flowtime-3

텍스트이(가) 표시된 사진

자동 생성된 설명

그 결과는 아래와 같다.

텍스트, 스크린샷, 노트북, 화면이(가) 표시된 사진

자동 생성된 설명

3번째 경우, 즉 **add an extra order fulfillment operator** 의 경우에서 가장 큰 overall flowtime reduce가 이루어진다.

**<5>**

PAN에 적용할 수 있도록 variables와 capacity, schedules을 적절하게 이미 이전 문항에서 설정해주었다. 이에 곧바로 PAN으로 넘어갔다.

테이블이(가) 표시된 사진

자동 생성된 설명

위와 같이 문제의 상황에 맞게 process analyzer에서 scenario를 추가해주었다. 시행 결과는 아래와 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

- Overall flowtime minimize는 시나리오 5에서 이루어진다. extra sales operator과 extra order fulfillment operator를 각각 1명씩 고용하는 경우이다.

- Highest utilization은 시나리오 1에서 이루어진다. extra sales operator를 2명 고용하는 경우이다.

- Fewest number remaining은 시나리오 3에서 이루어진다. extra order fulfillment operator를 2명 고용하는 경우이다.

**<6>**

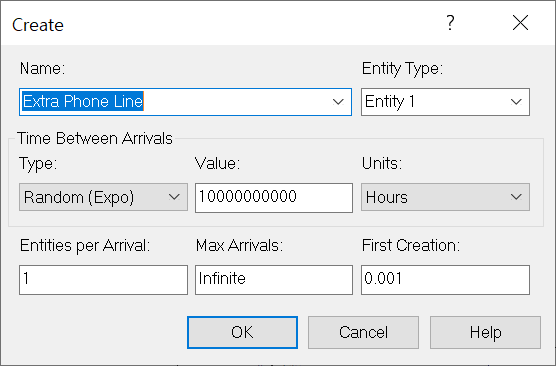
Extra carts, extra phone lines variable을 추가로 생성한 후, 이전과 같이 carts resource의 capacity 설정을 위해 cart schedule을 생성한 후 적용했다.

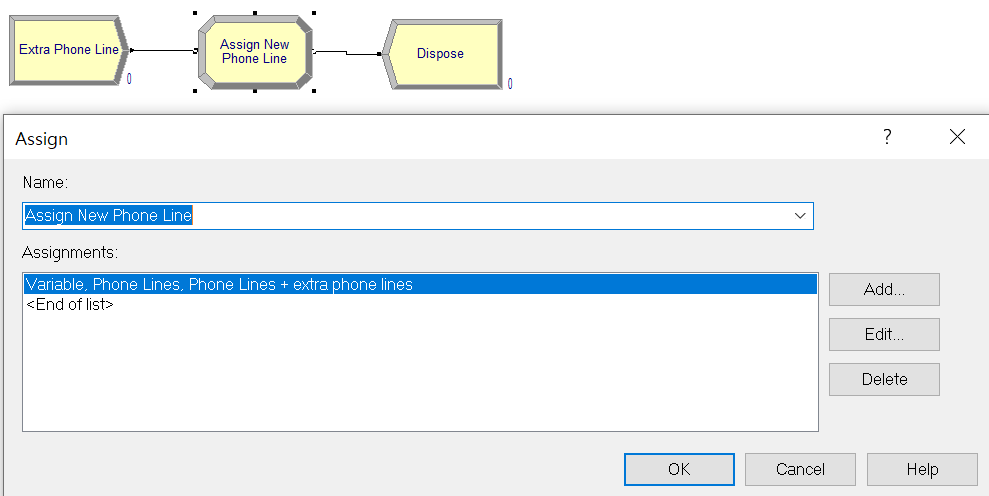


테이블이(가) 표시된 사진

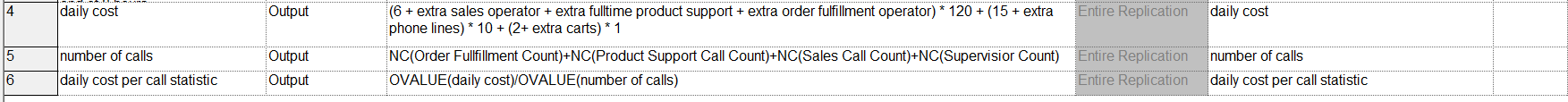
자동 생성된 설명

그런데, 문제에서 설명한 대로 phone lines는 resource가 아닌 variable로서 있기에 시작 직후에(시작 후 0.001시간 후) phone line variable의 값을 (기존 값 + extra phone lines)으로 할당해주는 부분을 모델링했다.





이후에, 문제에서 요구한 대로 daily cost, number of calls, daily cost per call statistic 을 정의했다.

 - daily cost

(6 + extra sales operator + extra fulltime product support + extra order fulfillment operator) \* 120 + (15 + extra phone lines) \* 10 + (2+ extra carts) \* 1

기존대로 위와 같이 설정하는 대신,

number of lines between 10 and 30 / number of carts between 1 and 10 이라는 제약 조건의 하한값을 설정하기 위하여

(6 + extra sales operator + extra fulltime product support + extra order fulfillment operator) \* 120 + (10 + extra phone lines) \* 10 + (1+ extra carts) \* 1

으로 설정한 뒤 extra phone lines과 extra carts의 범위를 알맞게 설정했다. 자세한 내용은 아래에서 다루겠다.

- number of calls

NC(Order Fullfillment Count)+NC(Product Support Call Count)+NC(Sales Call Count)+NC(Supervisior Count)

- daily cost per call statistic

daily cost / number of calls

이제, “optimal” plan을 찾기 위해 OptQuest를 이용했다.

- Objective

Minimize daily cost per call statistic



- Control

0 <= extra sales operator <= 10

0 <= extra fulltime product support <= 10

0 <= extra order fulfillment operator <= 10

0 <= extra phone lines <= 20 (phone lines default 10으로 설정)

0 <= extra carts <= 9 (carts default 1으로 설정)

텍스트, 스크린샷, 실내이(가) 표시된 사진

자동 생성된 설명

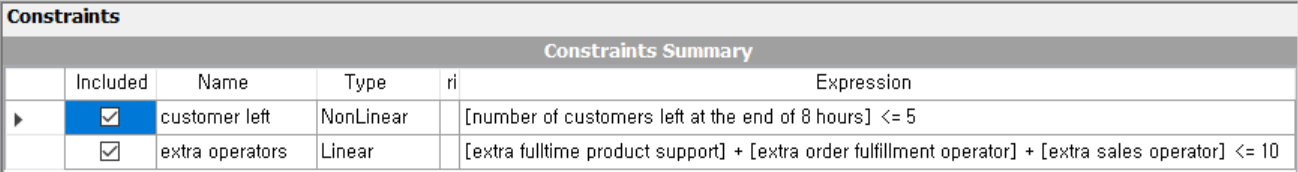
텍스트, 스크린샷, 실내이(가) 표시된 사진

자동 생성된 설명

- Constraints

Extra operators <= 10

number of phone call left over at the end of the day <= 5



- Optimization setting

use between 3 and 10 replications

stopping when the half-width is within 5% of the mean

테이블이(가) 표시된 사진

자동 생성된 설명

- 최종 결과 Optimal Plan

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

즉, carts 3개 / Phone Lines 11개 / Extra sales operator 0명 / extra product support operator 2명 / extra order fullfillment operator 1명 일 때가 최적의 plan이고,

이 때의 **daily cost per call = 2.287747** 이 최적이다.

테이블이(가) 표시된 사진

자동 생성된 설명

위 plan은 추가 고용자 3명, number of customers left 4.1명으로 제약 조건을 만족시키는 것을 확인할 수 있다.